



AWS ParallelCluster Guida per l'utente (v3)

AWS ParallelCluster



AWS ParallelCluster: AWS ParallelCluster Guida per l'utente (v3)

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è AWS ParallelCluster	1
Come AWS ParallelCluster funziona	2
AWS ParallelCluster processi	2
clustermgtd	2
clusterstatusmgtd	3
computemgtd	3
AWS servizi usati da AWS ParallelCluster	4
Gateway Amazon API	5
AWS Batch	5
CloudFormation	5
Amazon CloudWatch	6
CloudWatch Eventi Amazon	6
CloudWatch Registri Amazon	6
AWS CodeBuild	6
Amazon DynamoDB	7
Amazon Elastic Block Store	7
Amazon Elastic Compute Cloud	7
Amazon Elastic Container Registry	8
Amazon EFS	8
Amazon FSx per Lustre	8
Amazon FSx per NetApp ONTAP	8
Amazon FSx per OpenZFS	9
AWS Identity and Access Management	9
AWS Lambda	9
Amazon RDS	9
Amazon Route 53	10
Amazon Simple Notification Service	10
Amazon Simple Storage Service	10
Amazon VPC	10
Elastic Fabric Adapter	11
EC2 Image Builder	11
Amazon DCV	11
AWS ParallelCluster elenchi interni	11
Configurazione AWS ParallelCluster	13

Prerequisiti	13
Configurare un Account AWS	13
Creazione di una coppia di chiavi	13
Installazione della AWS ParallelCluster CLI	14
Installa AWS ParallelCluster in un ambiente virtuale (consigliato)	14
Installazione AWS ParallelCluster in un ambiente non virtuale usando pip	16
Installa AWS ParallelCluster come applicazione autonoma	17
Passaggi da eseguire dopo l'installazione	19
Installazione dell'interfaccia utente AWS ParallelCluster	20
Installa la PCUI	20
Parametri dello stack	23
Configura un dominio personalizzato	27
Opzioni del pool di utenti di Amazon Cognito	29
Identifica la versione e la versione PCUI AWS ParallelCluster	31
Costi PCUI	32
Nozioni di base	32
Configura e crea un cluster con la AWS ParallelCluster CLI	33
Configura e crea un cluster con l'interfaccia utente AWS ParallelCluster	43
Connessione a un cluster	45
Accesso di più utenti ai cluster	45
Crea un Active Directory	46
Crea un cluster con un dominio AD	47
Accedere a un cluster integrato con un dominio AD	50
Esecuzione di processi MPI	51
Esempio di configurazioni di cluster AWS Managed Microsoft AD LDAP (S)	51
Best practice	55
Procedure consigliate: selezione del tipo di istanza del nodo principale	56
Migliori pratiche: prestazioni di rete	56
Migliori pratiche: avvisi sul budget	58
Migliori pratiche: spostare un cluster in una nuova versione AWS ParallelCluster secondaria o patch	58
Passaggio da AWS ParallelCluster 2.x a 3.x	59
Azioni Bootstrap personalizzate	59
AWS ParallelCluster 2.x e 3.x utilizzano una sintassi diversa del file di configurazione	60
Lingua inclusiva	66
Supporto Scheduler	66

AWS ParallelCluster CLI	66
Aggiornamento della configurazione IMDS	69
Usando AWS ParallelCluster	70
AWS ParallelCluster INTERFACCIA UTENTE	71
AWS Lambda Configurazione VPC in AWS ParallelCluster	73
AWS Identity and Access Management autorizzazioni in AWS ParallelCluster	74
AWS ParallelCluster Ruoli delle istanze Amazon EC2	75
AWS ParallelCluster esempi di politiche <code>pcluster</code> utente	76
AWS ParallelCluster esempi di politiche utente per la gestione delle risorse IAM	91
AWS ParallelCluster parametri di configurazione per gestire le autorizzazioni IAM	98
Configurazioni di rete	113
AWS ParallelCluster in un'unica sottorete pubblica	114
AWS ParallelCluster utilizzando due sottoreti	115
AWS ParallelCluster in una singola sottorete privata connessa tramite AWS Direct Connect	117
AWS ParallelCluster con scheduler AWS Batch	118
AWS ParallelCluster in un'unica sottorete senza accesso a Internet	120
Nodi di accesso forniti da AWS ParallelCluster	127
Sicurezza per i nodi di accesso	127
Rete per i nodi di accesso	128
Archiviazione per i nodi di accesso	128
Proprietà <code>Imds</code> per i nodi di accesso	129
Ciclo di vita dei nodi di accesso	129
Autorizzazioni necessarie per eseguire il pool di nodi di accesso	130
Azioni bootstrap personalizzate	130
Configurazione	133
Arguments (Argomenti)	137
Cluster di esempio con azioni bootstrap personalizzate	137
Esempio di aggiornamento di uno script di bootstrap personalizzato per IMDSv2	139
Esempio di come aggiornare una configurazione per IMDSv1	139
Lavorare con Amazon S3	140
Esempi	140
Utilizzo di Istanze spot	141
Scenario 1: viene interrotta un'istanza Spot senza attività in esecuzione	142
Scenario 2: l'istanza spot che esegue attività a nodo singolo viene interrotta	142
Scenario 3: l'istanza spot che esegue attività a più nodi viene interrotta	142

Scheduler supportati da AWS ParallelCluster	143
Slurm Workload Manager	143
AWS Batch	209
Archiviazione condivisa	216
Amazon EBS	219
Amazon EFS	220
FSx per Lustre	221
FSx per ONTAP, FSx per OpenZFS e File Cache	222
Utilizzo dello storage condiviso	222
Quote	226
Assegnazione di tag	227
Visualizzare i tag	228
Monitoraggio AWS ParallelCluster e registri	230
Integrazione con Amazon CloudWatch Logs	231
CloudWatch Pannello di controllo Amazon	234
CloudWatch Allarmi Amazon per le metriche dei cluster	236
AWS ParallelCluster rotazione dei log configurata	238
pcclusterRegistri CLI	240
Registri di output della EC2 console Amazon	240
Recupera i log di PCUI e AWS ParallelCluster di runtime	241
Recupero e conservazione dei log	243
AWS CloudFormation risorsa personalizzata	246
Stack di provider ospitato da AWS ParallelCluster	247
Risorsa del cluster	249
Operazioni sui cluster	252
Risoluzione dei problemi relativi agli stack che includono la risorsa personalizzata AWS ParallelCluster	253
Elastic Fabric Adapter	253
Configurazione di rete EFA predefinita	254
Personalizzazione delle interfacce di rete EFA	254
Abilitare Intel MPI	255
AWS ParallelCluster API	256
AWS ParallelCluster Documentazione API	257
Implementa l'API con AWS ParallelCluster AWS CLI	258
Aggiornamento dell'API	261
Invocare l'API AWS ParallelCluster	261

Accesso ai log e alle metriche dell'API	264
AWS ParallelCluster per Terraform	264
Connect ai nodi principali e di accesso tramite Amazon DCV	265
Certificato HTTPS Amazon DCV	266
Licenze Amazon DCV	266
Uso di <code>pcluster update-cluster</code>	266
Politica di aggiornamento: definizioni	267
Esempi di <code>pcluster update-cluster</code>	272
AWS ParallelCluster Personalizzazione AMI	275
AWS ParallelCluster Considerazioni sulla personalizzazione delle AMI	275
Esegui test di convalida dei componenti personalizzati	276
Monitora il processo di Image Builder con <code>pcluster</code> comandi per facilitare il debug	276
Altre considerazioni	277
Avvio di istanze con On-Demand Capacity Reservations (ODCR)	278
Utilizzo di ODCR con AWS ParallelCluster	278
Avvia istanze con Capacity Blocks (CB)	287
Usare CB con AWS ParallelCluster	287
Patch AMI e sostituzione delle EC2 istanze Amazon	289
Aggiornamento o sostituzione dell'istanza del nodo principale	290
Salva i dati da unità temporanee	290
Arresta e avvia il nodo principale di un cluster	291
Sistemi operativi	293
Considerazioni sul sistema operativo	293
Riferimento per AWS ParallelCluster	295
AWS ParallelCluster comandi CLI versione 3	295
<code>pcluster</code>	296
<code>pcluster3-config-converter</code>	342
File di configurazione	343
File di configurazione del cluster	344
Crea file di configurazione delle immagini	492
AWS ParallelCluster Riferimento API	502
BuildImage	503
createCluster	508
Elimina cluster	514
deleteClusterInstances	517
Elimina immagine	519

Descrivi Cluster	522
describeClusterInstances	530
describeComputeFleet	534
Descrivi l'immagine	535
getClusterLogEventi	542
getClusterStackEventi	546
getImageLogEventi	550
getImageStackEventi	554
Elenca i cluster	558
listClusterLogStream	562
listImageLogStream	566
Elenca immagini	570
listOfficialImages	573
Aggiorna Cluster	576
updateComputeFleet	583
AWS ParallelCluster API della libreria Python	585
AWS ParallelCluster Autorizzazione della libreria Python	585
Installa la AWS ParallelCluster libreria Python	586
Operazioni dell'API del cluster	586
Elabora le operazioni API della flotta	590
Operazioni di cluster e stack log	592
Operazioni dell'API di immagine	595
Operazioni relative ai log delle immagini e degli stack	597
Esempio	600
AWS Lambda per la libreria AWS ParallelCluster Python	601
Esercitazioni	603
Esecuzione del tuo primo lavoro su AWS ParallelCluster	603
Verifica dell'installazione	604
Creazione del primo cluster	604
Accedere al nodo principale	605
Esecuzione del primo lavoro con Slurm	606
Creazione di un' AWS ParallelCluster AMI personalizzata	607
Come personalizzare l' AWS ParallelCluster AMI	608
Crea un' AWS ParallelCluster AMI personalizzata	608
Modificare un AWS ParallelCluster AMI	615
Integrazione di Active Directory	618

Crea l'infrastruttura AD	619
(Facoltativo) Gestisci utenti e gruppi AD	636
Creazione del cluster	638
Connect al cluster come utente	644
Eliminazione	645
Configurazione della crittografia dello storage condiviso con una chiave AWS KMS	650
Creare la policy	651
Configura e crea il cluster	652
Esecuzione di processi in un cluster in modalità coda multipla	653
Configura il tuo cluster	654
Creazione di un cluster	655
Accedere al nodo principale	656
Esegue il lavoro in modalità coda multipla	657
Utilizzo dell' AWS ParallelCluster API	661
Creazione di un cluster con Slurm contabilità	675
Fase 1: Creare il VPC e le sottoreti per AWS ParallelCluster	676
Fase 2: Creare lo stack di database	676
Fase 3: Creare un cluster con Slurm contabilità abilitata	676
Creazione di un cluster con un cluster esterno Slurmdbd contabilità	677
Passaggio 1: creare lo stack Slurmdbd	678
Fase 2: Creare un cluster con elementi esterni Slurmdbd abilitato	680
Ripristino a una versione precedente del documento AWS Systems Manager	681
Ripristina una versione precedente del documento SSM	681
Creazione di un cluster con CloudFormation	683
Creazione di cluster con uno CloudFormation stack di creazione rapida	684
Creazione di cluster con l'interfaccia CloudFormation a riga di comando (CLI)	686
Visualizza l'output del cluster CloudFormation	688
Accedi al tuo cluster	689
Eliminazione	689
Implementa l' ParallelCluster API con Terraform	690
Definisci un progetto Terraform	690
Distribuzione dell'API	692
Autorizzazioni richieste	693
Creare un cluster con Terraform	696
Definisci un progetto Terraform	696
Distribuzione del cluster	703

Autorizzazioni richieste	704
Creazione di un'AMI personalizzata con Terraform	705
Definisci un progetto Terraform	705
Implementa l'AMI	708
Autorizzazioni richieste	709
AWS ParallelCluster Integrazione dell'interfaccia utente con Identity Center	710
Abilita IAM Identity Center	710
Aggiungere l'applicazione a IAM Identity Center	712
Esecuzione di lavori containerizzati con Pyxis	715
Creazione del cluster	716
Invia offerte di lavoro	718
Creazione di un cluster con un Lustre abilitato per EFA FSx	718
Requisiti	718
Creazione di gruppi di sicurezza	719
Crea il file system	720
Creazione del cluster	721
La convalida FSx con EFA funziona	722
Supporta Nvidia-IMEX con istanza p6e-gb200	723
Crea uno script Prolog per gestire Nvidia-IMEX	724
Crea lo script di HeadNode OnNodeStart azione personalizzato	728
Creazione del cluster	728
Convalida la configurazione IMEX	729
Personalizza le interfacce di rete dei nodi di calcolo con sostituzioni dei modelli di avvio	731
Fase 1: Creare gruppi di sicurezza	732
Fase 2: Creare il modello di lancio	733
Passaggio 3: Creare il cluster con le sostituzioni del modello di avvio	735
AWS ParallelCluster risoluzione dei problemi	736
Sto cercando di creare un cluster	737
failureCode è OnNodeConfiguredExecutionFailure	737
failureCode è OnNodeConfiguredDownloadFailure	738
failureCode è OnNodeConfiguredFailure	738
failureCode è OnNodeStartExecutionFailure	738
failureCode è OnNodeStartDownloadFailure	739
failureCode è OnNodeStartFailure	739
failureCode è EbsMountFailure	740
failureCode è EfsMountFailure	740

failureCode è FsxMountFailure	740
failureCode è RaidMountFailure	741
failureCode è AmiVersionMismatch	741
failureCode è InvalidAmi	741
failureCode è HeadNodeBootstrapFailure con failureReason Failed to setup the head node.	742
failureCode è scaduto HeadNodeBootstrapFailure il timeout per la creazione del failureReason cluster.	742
failureCode è HeadNodeBootstrapFailure con failureReason Failed to bootstrap the head node.	744
failureCode è ResourceCreationFailure	744
failureCode è ClusterCreationFailure	744
Vedere WaitCondition timed out... in CloudFormation pila	745
Vedere Resource creation cancelled in pila CloudFormation	745
Visualizzazione Failed to run cfn-init... o altri errori nello stack CloudFormation	745
La visione chef-client.log finisce con INFO: Waiting for static fleet capacity provisioning	745
Vedendo Failed to run preinstall or postinstall in cfn-init.log	745
Visualizzazione This AMI was created with xxx, but is trying to be used with xxx... in CloudFormation pila	745
Vedere This AMI was not baked by AWS ParallelCluster... in pila CloudFormation	745
Il pcluster create-cluster comando Seeing non viene eseguito localmente	746
Supporto aggiuntivo	746
Sto cercando di eseguire un lavoro	746
srun il processo interattivo fallisce e restituisce un errore srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf	746
Job è bloccato nello CF stato con squeue il comando	747
Esecuzione di lavori su larga scala e visualizzazione nfsd: too many open connections, consider increasing the number of threads in /var/log/messages	747
Esecuzione di un job MPI	748
Tentativo di aggiornamento di un cluster	748
pcluster update-cluster il comando non viene eseguito localmente	749

Vedere <code>clusterStatus</code> avviene <code>UPDATE_FAILED</code> tramite <code>pcluster describe-cluster</code> comando	749
L'aggiornamento del cluster è scaduto	750
Tentativo di accesso allo storage	750
Utilizzo di un file system Amazon FSx for Lustre esterno	750
Utilizzo di un file system Amazon Elastic File System esterno	750
Tentativo di eliminazione di un cluster	751
Il <code>pcluster delete-cluster</code> comando non viene eseguito localmente	751
Lo stack del cluster non viene eliminato	751
Sto cercando di aggiornare lo AWS ParallelCluster stack di API	751
Visualizzazione degli errori nelle inizializzazioni dei nodi di calcolo	751
Vedere in Node <code>bootstrap error clustermgtd.log</code>	752
Ho configurato le prenotazioni di capacità su richiesta (ODCRs) o le istanze riservate zonali	752
An error occurred (VcpuLimitExceeded)Attivo <code>slurm_resume.log</code> quando non riesco a eseguire un processo o attivo quando non riesco a creare un cluster <code>clustermgtd.log</code>	753
An error occurred (InsufficientInstanceCapacity)Attivo <code>slurm_resume.log</code> quando non riesco a eseguire un processo o attivo quando non riesco a creare un cluster <code>clustermgtd.log</code>	754
I nodi di visualizzazione sono DOWN in stato con Reason (Code:InsufficientInstanceCapacity)... ..	754
Vedere dentro <code>cannot change locale (en_US.utf-8) because it has an invalid name slurm_resume.log</code>	754
Nessuno degli scenari precedenti si applica alla mia situazione	755
Risoluzione dei problemi relativi alle metriche di integrità del cluster	755
Visualizzazione del grafico degli errori di provisioning delle istanze	755
Visualizzazione del grafico Unhealthy Instance Errors	757
Visualizzazione del grafico Compute Fleet Idle Time	759
Risoluzione dei problemi di distribuzione dei cluster	760
Visualizza CloudFormation gli eventi su <code>CREATE_FAILED</code>	761
Usa la CLI per visualizzare i flussi di log	763
Ricrea il cluster fallito con <code>rollback-on-failure</code>	765
Risoluzione dei problemi di distribuzione del cluster utilizzando Terraform	766
ParallelCluster API non trovata	766
Utente non autorizzato a chiamare l'API ParallelCluster	767

Risoluzione dei problemi di scalabilità	767
Registri chiave per il debug	768
Viene visualizzato un <code>InsufficientInstanceCapacity</code> errore <code>slurm_resume.log</code> quando non riesco a eseguire un processo o <code>clustermgtd.log</code> quando non riesco a creare un cluster	770
Risoluzione dei problemi di inizializzazione dei nodi	771
Risoluzione dei problemi di sostituzioni e terminazioni impreviste dei nodi	773
Sostituzione, interruzione o spegnimento di istanze e nodi problematici	775
Stato della coda (partizione) <code>Inactive</code>	776
Risoluzione di altri problemi noti relativi a nodi e processi	776
Gruppi di posizionamento e problemi di avvio delle istanze	776
Sostituzione delle directory	777
Risoluzione dei problemi in Amazon DCV	777
Registri per Amazon DCV	777
Problemi con Ubuntu Amazon DCV	777
Risoluzione dei problemi nei cluster con integrazione AWS Batch	778
Problemi relativi al nodo principale	778
Problemi di calcolo	778
Errori del processo	779
Errore <code>Connect timeout</code> sull'URL dell'endpoint	779
Risoluzione dei problemi di integrazione multiutente con Active Directory	779
Risoluzione dei problemi specifici di Active Directory	780
Abilita la modalità di debug	781
Come passare da LDAPS a LDAP	781
Come disabilitare la verifica dei certificati del server LDAPS	781
Come accedere con una chiave SSH anziché una password	782
Come reimpostare una password utente e le password scadute	782
Come verificare il dominio aggiunto	783
Come risolvere i problemi relativi ai certificati	783
Come verificare che l'integrazione con Active Directory funzioni	785
Come risolvere i problemi di accesso ai nodi di calcolo	786
Problemi noti con i job SimCenter StarCCM+ in un ambiente multiutente	786
Problemi noti relativi alla risoluzione dei nomi utente	787
Come risolvere i problemi di creazione della home directory	787
Risoluzione dei problemi relativi alle AMI personalizzate	788

Risoluzione dei problemi di un timeout di aggiornamento del cluster quando cfn-hup non è in esecuzione	789
Risoluzione dei problemi di rete	790
Problemi relativi al cluster in un'unica sottorete pubblica	790
Aggiornamento del cluster non riuscito a causa di un'azione personalizzata onNodeUpdated ..	790
Visualizzazione degli errori con la configurazione personalizzata Slurm	791
Allarmi cluster	791
Risoluzione delle modifiche alla configurazione del sistema operativo che causano errori o guasti	792
Problemi comuni di configurazione del sistema operativo	792
Le migliori pratiche per le modifiche alla configurazione del sistema operativo	796
AWS ParallelCluster politica di supporto	798
Sicurezza	800
Informazioni di sicurezza per i servizi utilizzati da AWS ParallelCluster	801
Protezione dei dati	801
Crittografia dei dati	802
Consulta anche	803
Identity and Access Management	804
Convalida della conformità	804
Applicazione di TLS 1.2	805
Determinare i protocolli attualmente supportati	806
Compilare OpenSSL e Python	807
Configurazione di gruppi di sicurezza per ambienti con restrizioni	808
Panoramica dei gruppi di sicurezza	809
Porte necessarie per il funzionamento del cluster	809
Creazione di gruppi di sicurezza personalizzati	811
Configurazione dei gruppi di sicurezza nella configurazione del cluster	812
Utilizzo degli endpoint VPC in ambienti con restrizioni	813
Le migliori pratiche per la configurazione dei gruppi di sicurezza	814
Risoluzione dei problemi relativi ai gruppi di sicurezza	814
Supportato Regioni AWS	816
Note di rilascio e cronologia dei documenti	818
.....	cmlxv

Che cos'è AWS ParallelCluster

AWS ParallelCluster è uno strumento di gestione dei cluster open source AWS supportato che consente di distribuire e gestire cluster HPC (High-Performance Computing) in Cloud AWS. Configura automaticamente le risorse di elaborazione, lo scheduler e il file system condiviso necessari. È possibile utilizzare con e AWS ParallelCluster AWS Batch Slurm pianificatori.

Con AWS ParallelCluster, puoi creare e implementare rapidamente ambienti di elaborazione HPC proof of concept e di produzione. Inoltre, puoi creare e implementare un flusso di lavoro di alto livello AWS ParallelCluster, come un portale di genomica che automatizza un intero flusso di lavoro di sequenziamento del DNA.

È possibile accedere utilizzando questi metodi: AWS ParallelCluster

- [AWS ParallelCluster interfaccia a riga di comando \(CLI\)](#)
- [AWS ParallelCluster API](#)
- [PCUI](#) (aggiunto con la versione 3.5.0)
- [AWS ParallelCluster API della libreria Python](#) (aggiunto con la versione 3.5.0)
- Come [AWS CloudFormation risorsa personalizzata](#) (aggiunto con la versione 3.6.0)

Prezzi

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Come AWS ParallelCluster funziona

AWS ParallelCluster è stato creato non solo come modo per gestire i cluster, ma come riferimento su come utilizzare AWS i servizi per creare un ambiente HPC. I seguenti argomenti descrivono AWS ParallelCluster i processi, Servizi AWS gli AWS ParallelCluster utenti e le modalità di utilizzo e le directory interne.

Argomenti

- [AWS ParallelCluster processi](#)
- [AWS servizi usati da AWS ParallelCluster](#)
- [AWS ParallelCluster elenchi interni](#)

AWS ParallelCluster processi

Questa sezione si applica ai cluster distribuiti con Slurm. Se utilizzato con questo scheduler, AWS ParallelCluster interagisce con il job scheduler sottostante per gestire il provisioning e la rimozione dei nodi di calcolo.

Per i cluster HPC basati su AWS Batch, si basa sulle funzionalità fornite da per AWS ParallelCluster gestire i nodi di elaborazione. AWS Batch

clustermgtd

Il daemon di gestione del cluster (`clustermgtd`) esegue le seguenti attività:

- Pulisci le partizioni inattive
- Gestisci Slurm le prenotazioni e i nodi associati ai Capacity Blocks (vedi la sezione seguente)
- Gestisci la capacità statica per assicurarti che sia sempre attiva e integra
- Sincronizza lo scheduler con Amazon EC2.
- Pulisci le istanze orfane
- Ripristina lo stato del nodo di pianificazione in caso di interruzione di Amazon EC2 che avviene al di fuori del flusso di lavoro di sospensione
- Gestisci le istanze Amazon EC2 non integre (quelle che non superano i controlli di integrità di Amazon EC2)
- Gestisci gli eventi di manutenzione programmata

- Gestisci i nodi di pianificazione non integri (quelli che non superano i controlli di integrità dello scheduler)

Gestione delle Slurm prenotazioni e dei nodi associati ai Capacity Blocks

ParallelCluster supporta On-Demand Capacity Reservations (ODCR) e Capacity Blocks for Machine Learning (CB). A differenza dell'ODCR, CB può avere orari di inizio futuri ed è limitato nel tempo.

`clustermgtd` cerca i nodi non integri in un ciclo, termina tutte le istanze Amazon EC2 non funzionanti e le sostituisce con nuove istanze se si tratta di nodi statici.

AWS ParallelCluster gestisce i nodi statici associati a Capacity Blocks in modo diverso: crea un cluster anche se il CB non è ancora attivo e avvia automaticamente le istanze una volta che il CB è attivo.

I Slurm nodi che corrispondono alle risorse di calcolo associate CBs a quelle non ancora attive vengono mantenuti nello stato di manutenzione fino al raggiungimento dell'ora di avvio del CB. Questi Slurm nodi rimangono in uno reservation/maintenance stato associato all'utente Slurm amministratore, il che significa che possono accettare lavori, ma i lavori rimangono in sospeso fino alla rimozione della Slurm prenotazione.

`clustermgtd` crea o elimina automaticamente Slurm le prenotazioni: mette i relativi nodi CB in uno stato di manutenzione basato sullo stato CB. Quando il CB diventa attivo, la Slurm prenotazione viene rimossa, i nodi si avviano e diventano disponibili per i lavori in sospeso o per l'invio di nuovi lavori.

Quando viene raggiunta l'ora di fine del CB, i nodi vengono riportati a uno stato. reservation/maintenance Spetta agli utenti eseguire resubmit/requeue il job su una nuova queue/compute risorsa quando il CB non è più attivo e le istanze vengono terminate.

clusterstatusmgtd

Il cluster status management daemon (`clusterstatusmgtd`) gestisce l'aggiornamento dello stato della flotta di calcolo. Ogni minuto recupera lo stato della flotta memorizzato in una tabella DynamoDB e gestisce qualsiasi richiesta. STOP/START

computemgtd

I processi di gestione del calcolo daemon (`computemgtd`) vengono eseguiti su ciascuno dei nodi di calcolo del cluster. Ogni cinque (5) minuti, il demone di gestione del calcolo conferma che il

nodo principale è raggiungibile ed è integro. Se trascorrono cinque (5) minuti durante i quali il nodo principale non può essere raggiunto o non è integro, il nodo di elaborazione viene spento.

AWS servizi usati da AWS ParallelCluster

I seguenti servizi Amazon Web Services (AWS) vengono utilizzati da AWS ParallelCluster.

Argomenti

- [Gateway Amazon API](#)
- [AWS Batch](#)
- [CloudFormation](#)
- [Amazon CloudWatch](#)
- [CloudWatch Eventi Amazon](#)
- [CloudWatch Registri Amazon](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx per Lustre](#)
- [Amazon FSx per NetApp ONTAP](#)
- [Amazon FSx per OpenZFS](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon RDS](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service](#)
- [Amazon Simple Storage Service](#)
- [Amazon VPC](#)
- [Elastic Fabric Adapter](#)

- [EC2 Image Builder](#)
- [Amazon DCV](#)

Gateway Amazon API

Amazon API Gateway è un AWS servizio che consente di creare, pubblicare, mantenere, monitorare e proteggere REST, HTTP e WebSocket APIs su qualsiasi scala

AWS ParallelCluster utilizza API Gateway per ospitare l' AWS ParallelCluster API.

Per ulteriori informazioni su Amazon API Gateway, consulta <https://aws.amazon.com/api-gateway/>e <https://docs.aws.amazon.com/apigateway/>.

AWS Batch

AWS Batch è un servizio AWS gestito di pianificazione dei lavori. Fornisce dinamicamente la quantità e il tipo ottimali di risorse di elaborazione (ad esempio, CPU o istanze ottimizzate per la memoria) nei cluster. AWS Batch Il provisioning di queste risorse viene effettuato in base ai requisiti specifici dei processi in batch, inclusi i requisiti di volume. Con AWS Batch, non è necessario installare o gestire software di elaborazione in batch o cluster di server aggiuntivi per eseguire i lavori in modo efficace.

AWS Batch viene utilizzato solo con i AWS Batch cluster.

Per ulteriori informazioni su AWS Batch, vedere <https://aws.amazon.com/batch/>e <https://docs.aws.amazon.com/batch/>.

CloudFormation

CloudFormation è un infrastructure-as-code servizio che fornisce un linguaggio comune per modellare AWS e fornire risorse applicative di terze parti nell'ambiente cloud. È il servizio principale utilizzato da AWS ParallelCluster. Ogni cluster in AWS ParallelCluster è rappresentato come uno stack e tutte le risorse richieste da ogni cluster sono definite all'interno del AWS ParallelCluster CloudFormation modello. Nella maggior parte dei casi, i comandi AWS ParallelCluster CLI corrispondono direttamente ai comandi CloudFormation dello stack, come create, update ed delete. Le istanze avviate all'interno di un cluster effettuano chiamate HTTPS all' CloudFormation endpoint in Regione AWS cui viene avviato il cluster.

Per ulteriori informazioni su CloudFormation, vedere <https://aws.amazon.com/cloudformation/>e <https://docs.aws.amazon.com/cloudformation/>

Amazon CloudWatch

Amazon CloudWatch (CloudWatch) è un servizio di monitoraggio e osservabilità che fornisce dati e approfondimenti utilizzabili. Queste informazioni possono essere utilizzate per monitorare le applicazioni, rispondere ai cambiamenti delle prestazioni e alle eccezioni dei servizi e ottimizzare l'utilizzo delle risorse. In AWS ParallelCluster, CloudWatch viene utilizzato per una dashboard, per monitorare e registrare le fasi di creazione dell'immagine Docker e l'output dei lavori. AWS Batch

Prima della AWS ParallelCluster versione 2.10.0, CloudWatch veniva utilizzato solo con i cluster. AWS Batch

Per ulteriori informazioni su CloudWatch, vedere e. <https://aws.amazon.com/cloudwatch/> <https://docs.aws.amazon.com/cloudwatch/>

CloudWatch Eventi Amazon

Amazon CloudWatch Events (CloudWatch Events) offre un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle risorse di Amazon Web Services (AWS). Utilizzando semplici regole che puoi impostare rapidamente, puoi abbinare eventi e instradarli verso una o più funzioni o flussi del target. In AWS ParallelCluster, CloudWatch Events viene utilizzato per AWS Batch lavori.

Per ulteriori informazioni sugli CloudWatch eventi, vedere <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cwe-now-eb>.

CloudWatch Registri Amazon

Amazon CloudWatch Logs (CloudWatch Logs) è una delle funzionalità principali di Amazon. CloudWatch Puoi usarlo per monitorare, archiviare, visualizzare e cercare nei file di registro molti dei componenti utilizzati da. AWS ParallelCluster

Prima della AWS ParallelCluster versione 2.6.0, CloudWatch Logs veniva utilizzato solo con i cluster. AWS Batch

Per ulteriori informazioni, consulta [Integrazione con Amazon CloudWatch Logs](#).

AWS CodeBuild

AWS CodeBuild (CodeBuild) è un servizio di integrazione continua AWS gestito che compila il codice sorgente, esegue test e produce pacchetti software pronti per l'implementazione. In AWS

ParallelCluster, CodeBuild viene utilizzato per creare immagini Docker in modo automatico e trasparente quando vengono creati i cluster.

CodeBuild viene utilizzato solo con i cluster. AWS Batch

Per ulteriori informazioni su CodeBuild, vedere <https://aws.amazon.com/codebuild/>e <https://docs.aws.amazon.com/codebuild/>.

Amazon DynamoDB

Amazon DynamoDB (DynamoDB) è un servizio di database NoSQL veloce e flessibile. Viene utilizzato per archiviare le informazioni minime sullo stato del cluster. Il nodo principale tiene traccia delle istanze assegnate in una tabella DynamoDB.

DynamoDB non viene utilizzato con i cluster. AWS Batch

Per ulteriori informazioni su DynamoDB, vedere e. <https://aws.amazon.com/dynamodb/><https://docs.aws.amazon.com/dynamodb/>

Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) Elastic Block Store (Amazon EBS) è un servizio di storage a blocchi ad alte prestazioni che fornisce storage persistente per volumi condivisi. Tutte le impostazioni di Amazon EBS possono essere passate attraverso la configurazione. I volumi Amazon EBS possono essere inizializzati vuoti o da uno snapshot Amazon EBS esistente.

Per ulteriori informazioni su Amazon EBS, consulta <https://aws.amazon.com/ebs/>e <https://docs.aws.amazon.com/ebs/>.

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) fornisce la capacità di elaborazione per AWS ParallelCluster. I nodi principali e di calcolo sono EC2 istanze Amazon. È possibile selezionare qualsiasi tipo di istanza che supporti la macchina virtuale hardware (HVM). I nodi head e compute possono essere di diversi tipi di istanza. Inoltre, se vengono utilizzate più code, alcuni o tutti i nodi di calcolo possono essere avviati anche come istanza Spot. I volumi di Instance Store trovati sulle istanze vengono montati come volumi LVM (Logical Volume Manager) con striping.

Per ulteriori informazioni su Amazon EC2 , consulta <https://aws.amazon.com/ec2/>e <https://docs.aws.amazon.com/ec2/>.

Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) è un registro di container Docker completamente gestito che semplifica l'archiviazione, la gestione e la distribuzione di immagini di container Docker. In AWS ParallelCluster, Amazon ECR archivia le immagini Docker create quando vengono creati i cluster. Le immagini Docker vengono quindi utilizzate da per eseguire AWS Batch i contenitori per i lavori inviati.

Amazon ECR viene utilizzato solo con i AWS Batch cluster.

Per ulteriori informazioni, consultare <https://aws.amazon.com/ecr/> e <https://docs.aws.amazon.com/ecr/>.

Amazon EFS

Amazon Elastic File System (Amazon EFS) fornisce un file system NFS elastico semplice, scalabile e completamente gestito da utilizzare con Cloud AWS servizi e risorse locali. Amazon EFS viene utilizzato quando [EfsSettings](#) sono specificati. Il supporto per Amazon EFS è stato aggiunto nella AWS ParallelCluster versione 2.1.0.

Per ulteriori informazioni su Amazon EFS, consulta <https://aws.amazon.com/efs/> e <https://docs.aws.amazon.com/efs/>.

Amazon FSx per Lustre

FSx for Lustre fornisce un file system ad alte prestazioni che utilizza il file system Lustre open source. FSx for Lustre viene utilizzato quando sono specificati. [Proprietà FsxLustreSettings](#) Il supporto FSx per Lustre è stato aggiunto nella AWS ParallelCluster versione 2.2.1.

[Per ulteriori informazioni su FSx Lustre, vedere https://aws.amazon.com/fsx/ lustre/ and. https://docs.aws.amazon.com/fsx/](#)

Amazon FSx per NetApp ONTAP

FSx for ONTAP fornisce un sistema di storage condiviso completamente gestito basato sul NetApp popolare file system ONTAP. FSx for ONTAP viene utilizzato quando [Proprietà FsxOntapSettings](#) specificato. Il supporto FSx per ONTAP è stato aggiunto nella AWS ParallelCluster versione 3.2.0.

[Per ulteriori informazioni su ONTAP, vedere FSx https://aws.amazon.com/fsx/ netapp-ontap/ and. https://docs.aws.amazon.com/fsx/](#)

Amazon FSx per OpenZFS

FSx per OpenZFS fornisce un sistema di storage condiviso completamente gestito basato sul popolare file system OpenZFS. FSx for OpenZFS viene utilizzato quando sono specificati. [Proprietà FsxOpenZfsSettings](#) Il supporto FSx per OpenZFS è stato aggiunto nella AWS ParallelCluster versione 3.2.0.

[Per ulteriori informazioni su OpenZFS, FSx vedere openzfs/ and. https://aws.amazon.com/fsx/ https://docs.aws.amazon.com/fsx/](#)

AWS Identity and Access Management

AWS Identity and Access Management (IAM) viene utilizzato AWS ParallelCluster all'interno per fornire un ruolo IAM con privilegi minimi EC2 per Amazon per l'istanza specifica per ogni singolo cluster. AWS ParallelCluster alle istanze viene concesso l'accesso solo alle chiamate API specifiche necessarie per distribuire e gestire il cluster.

Con AWS Batch i cluster, i ruoli IAM vengono creati anche per i componenti coinvolti nel processo di creazione delle immagini Docker al momento della creazione dei cluster. Questi componenti includono le funzioni Lambda che possono aggiungere ed eliminare immagini Docker da e verso il repository Amazon ECR. Includono anche le funzioni consentite per eliminare il bucket Amazon S3 creato per il cluster e il progetto. CodeBuild Esistono anche ruoli per AWS Batch risorse, istanze e lavori.

Per ulteriori informazioni su IAM, consulta <https://aws.amazon.com/iam/> e <https://docs.aws.amazon.com/iam/>.

AWS Lambda

AWS Lambda (Lambda) esegue le funzioni che orchestrano la creazione di immagini Docker. Lambda gestisce anche la pulizia delle risorse cluster personalizzate, come le immagini Docker archiviate nel repository Amazon ECR e su Amazon S3.

Per ulteriori informazioni su Lambda, vedere <https://aws.amazon.com/lambda/> e <https://docs.aws.amazon.com/lambda/>

Amazon RDS

Amazon Relational Database Service (Amazon RDS) è un servizio Web che semplifica la configurazione, il funzionamento e la scalabilità di un database relazionale nel AWS cloud.

AWS ParallelCluster utilizza Amazon RDS per AWS Batch eSlurm.

Per ulteriori informazioni su Amazon RDS, consulta <https://aws.amazon.com/rds/> e <https://docs.aws.amazon.com/rds/>.

Amazon Route 53

Amazon Route 53 (Route 53) viene utilizzato per creare zone ospitate con nomi host e nomi di dominio completi per ciascuno dei nodi di elaborazione.

Per ulteriori informazioni su Route 53, consulta <https://aws.amazon.com/route53/> e <https://docs.aws.amazon.com/route53/>.

Amazon Simple Notification Service

(Amazon SNS) è un servizio gestito che fornisce il recapito dei messaggi dagli editori agli abbonati (noti anche come produttori e consumatori).

AWS ParallelCluster utilizza Amazon SNS per l'hosting delle API.

Per ulteriori informazioni su Amazon SNS, consulta <https://aws.amazon.com/sns/> e <https://docs.aws.amazon.com/sns/>.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) AWS ParallelCluster archivia i modelli che si trovano in ciascuno di essi. Regione AWS AWS ParallelCluster può essere configurato per consentire agli CLI/SDK strumenti di utilizzare Amazon S3.

AWS ParallelCluster crea anche un bucket Amazon S3 Account AWS per archiviare le risorse utilizzate dai cluster, come il file di configurazione del cluster. AWS ParallelCluster mantiene un bucket Amazon S3 in ogni bucket in Regione AWS cui crei i cluster.

Quando usi il AWS Batch cluster, un bucket Amazon S3 nel tuo account viene utilizzato per archiviare i dati correlati. Ad esempio, il bucket memorizza gli artefatti creati quando un'immagine Docker e gli script vengono creati a partire dai lavori inviati.

Per ulteriori informazioni, consultare <https://aws.amazon.com/s3/> e <https://docs.aws.amazon.com/s3/>.

Amazon VPC

Un Amazon Virtual Private Cloud (VPC) definisce una rete utilizzata dai nodi del cluster.

Per ulteriori informazioni su Amazon VPC, consulta <https://aws.amazon.com/vpc/>e. <https://docs.aws.amazon.com/vpc/>

Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) è un'interfaccia di rete per istanze che è possibile utilizzare per eseguire applicazioni che richiedono alti livelli di comunicazioni tra nodi su larga scala. AWS

Per ulteriori informazioni su Elastic Fabric Adapter, consulta [efa/](https://aws.amazon.com/hpc/efa/). [https://aws.amazon.com/hpc/](https://aws.amazon.com/hpc/efa/)

EC2 Image Builder

EC2 Image Builder è un AWS servizio completamente gestito che consente di automatizzare la creazione, la gestione e l'implementazione di immagini server personalizzate, sicure e up-to-date.

AWS ParallelCluster utilizza Image Builder per creare e gestire AWS ParallelCluster immagini.

Per ulteriori informazioni su EC2 Image Builder, vedere <https://aws.amazon.com/image-builder/>e. <https://docs.aws.amazon.com/imagebuilder/>

Amazon DCV

Amazon DCV è un protocollo di visualizzazione remota ad alte prestazioni che offre un modo sicuro per distribuire desktop remoti e lo streaming di applicazioni a qualsiasi dispositivo in condizioni di rete diverse. Amazon DCV viene utilizzato quando vengono specificate [Dcv](#) le impostazioni [Sezione HeadNode/](#). Il supporto per Amazon DCV è stato aggiunto nella AWS ParallelCluster versione 2.5.0.

Per ulteriori informazioni su Amazon DCV, consulta <https://aws.amazon.com/hpc/dcv/> and. <https://docs.aws.amazon.com/dcv/>

AWS ParallelCluster elenchi interni

Esistono diverse directory interne che vengono AWS ParallelCluster utilizzate per condividere i dati all'interno del cluster. Le seguenti directory sono condivise tra il nodo principale, i nodi di calcolo e i nodi di accesso:

- /opt/slurm
- /opt/intel
- /opt/parallelcluster/shared (only with compute nodes)

- `/opt/parallelcluster/shared_login_nodes` (only with login nodes)
- `/home` (unless specified in `SharedStorage`)

Note

Per impostazione predefinita, queste directory vengono create sul volume EBS del nodo principale e condivise come esportazioni NFS verso i nodi di calcolo e di accesso. A partire dalla AWS ParallelCluster versione 3.8, puoi AWS ParallelCluster abilitare la creazione e la gestione di un file system Amazon EFS per ospitare e condividere queste directory impostando il parametro su `efs`. [SharedStorageType](#)

Quando il cluster è ridimensionato, le esportazioni NFS tramite il volume EBS possono comportare problemi di prestazioni. Utilizzando EFS, è possibile evitare le esportazioni NFS man mano che il cluster si espande orizzontalmente ed evitare i colli di bottiglia a livello di prestazioni ad esse associati.

Configurazione AWS ParallelCluster

I seguenti argomenti descrivono come eseguire la configurazione. AWS ParallelCluster Imparerai come installare gli strumenti necessari e come utilizzarli, come implementare e gestire l'accesso di più utenti ai cluster e le migliori pratiche.

Argomenti

- [Prerequisiti](#)
- [Installazione di AWS ParallelCluster interfaccia a riga di comando \(CLI\)](#)
- [Passaggi da eseguire dopo l'installazione](#)
- [Installazione del PCUI](#)
- [Nozioni di base su AWS ParallelCluster](#)
- [Accesso di più utenti ai cluster](#)
- [Best practice](#)
- [Passaggio da AWS ParallelCluster 2.x a 3.x](#)

Prerequisiti

Prima di iniziare la configurazione e l'utilizzo AWS ParallelCluster, assicurati di aver completato i seguenti prerequisiti.

Configurare un Account AWS

Configura un AWS account da utilizzare AWS ParallelCluster.

Registrati per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida [introduttiva a un Account AWS](#) nella Guida Gestione dell'account AWS di riferimento.

Creazione di una coppia di chiavi

Per distribuire i cluster, AWS ParallelCluster avvia le istanze Amazon EC2 per creare il nodo principale del cluster e i nodi di calcolo. Per eseguire attività del cluster, come l'esecuzione e il

monitoraggio dei lavori o la gestione degli utenti, devi poter accedere al nodo principale del cluster. Per verificare di poter accedere all'istanza del nodo principale tramite SSH, devi utilizzare una coppia di chiavi Amazon EC2. Per informazioni su come creare una coppia di chiavi, consulta [Create a key pair](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Installazione di AWS ParallelCluster interfaccia a riga di comando (CLI)

AWS ParallelCluster è distribuito come pacchetto Python e viene installato utilizzando il gestore di pacchetti pip Python. Per istruzioni su come installare i pacchetti Python, consulta [Installazione dei pacchetti](#) nella Python Packaging User Guide.

Modi per l'installazione AWS ParallelCluster:

- [Installa AWS ParallelCluster in un ambiente virtuale \(consigliato\)](#)
- [Installazione AWS ParallelCluster in un ambiente non virtuale usando pip](#)
- [Installa AWS ParallelCluster come applicazione autonoma](#)

Puoi trovare il numero di versione della CLI più recente nella [pagina delle versioni](#) su GitHub. In questa guida, gli esempi di comandi presuppongono che tu abbia installato una versione di Python successiva alla versione 3.6. Gli esempi di comando pip utilizzano la versione pip3.

Gestisci entrambi AWS ParallelCluster 2 e AWS ParallelCluster 3

Se utilizzi sia AWS ParallelCluster 2 che AWS ParallelCluster 3 e desideri gestire le CLI per entrambi i pacchetti, ti consigliamo di installare AWS ParallelCluster 2 e AWS ParallelCluster 3 in [ambienti virtuali](#) diversi. In questo modo è possibile continuare a utilizzare ogni versione AWS ParallelCluster e tutte le risorse del cluster associate.

Installa AWS ParallelCluster in un ambiente virtuale (consigliato)

Si consiglia di eseguire l'installazione AWS ParallelCluster in un ambiente virtuale per evitare conflitti di versione dei requisiti con altri pip pacchetti.

Prerequisiti

- AWS ParallelCluster richiede Python 3.7 o successivo. [Se non lo hai già installato, scarica una versione compatibile per la tua piattaforma su python.org.](#)

Per installare AWS ParallelCluster in un ambiente virtuale

1. Se `virtualenv` non è installato, installalo utilizzando `pip3`. Se `python3 -m virtualenv help` visualizza le informazioni della Guida, andare alla fase 2.

```
$ python3 -m pip install --upgrade pip
$ python3 -m pip install --user --upgrade virtualenv
```

Eseguire `exit` per uscire dalla finestra corrente del terminale e aprirne una nuova per rilevare le modifiche apportate all'ambiente.

2. Creare un ambiente virtuale e assegnargli un nome.

```
$ python3 -m virtualenv ~/apc-ve
```

In alternativa, è possibile utilizzare l'opzione `-p` per specificare una determinata versione di Python.

```
$ python3 -m virtualenv -p $(which python3) ~/apc-ve
```

3. Attivare il nuovo ambiente virtuale.

```
$ source ~/apc-ve/bin/activate
```

4. Installa AWS ParallelCluster nel tuo ambiente virtuale.

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster"
```

5. Installa Node Version Manager e la versione più recente di Long-Term Support (LTS) Node.js . AWS Cloud Development Kit (AWS CDK) richiede Node.js CloudFormation per la generazione di modelli.

Note

Se Node.js l'installazione non funziona sulla tua piattaforma, puoi installare una versione LTS precedente alla versione LTS più recente. Per ulteriori informazioni, consulta il [programma di Node.js rilascio](#) e i prerequisiti [AWS CDK](#).

Esempio di comando Node.js di installazione:

```
$ nvm install --lts=Hydrogen
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

6. Verifica che AWS ParallelCluster sia installato correttamente.

```
$ pcluster version
{
  "version": "3.15.0"
}
```

Puoi utilizzare il comando `deactivate` per chiudere l'ambiente virtuale. Ogni volta che si avvia una sessione, è necessario [riattivare l'ambiente](#).

Per eseguire l'aggiornamento alla versione più recente di AWS ParallelCluster, esegui nuovamente il comando di installazione.

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster"
```

Installazione AWS ParallelCluster in un ambiente non virtuale usando pip

Puoi anche installarlo AWS ParallelCluster in un ambiente non virtuale usando pip, un gestore di pacchetti Python.

Prerequisiti

- AWS ParallelCluster richiede Python 3.7 o successivo. [Se non lo hai già installato, scarica una versione compatibile per la tua piattaforma su python.org](#).

Installa AWS ParallelCluster

1. Usa per installare pip. AWS ParallelCluster

```
$ python3 -m pip install "aws-parallelcluster" --upgrade --user
```

Quando si utilizza lo `--user` switch, pip AWS ParallelCluster installa `~/.local/bin` su.

2. Installa Node Version Manager e la versione più recente di Long-Term Support (LTS) Node.js . AWS Cloud Development Kit (AWS CDK) richiede Node.js CloudFormation per la generazione di modelli.

Note

Se Node.js l'installazione non funziona sulla tua piattaforma, puoi installare una versione LTS precedente alla versione LTS più recente. Per ulteriori informazioni, consulta il [programma di Node.js rilascio](#) e i prerequisiti [AWS CDK](#).

```
$ nvm install --lts=Hydrogen
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

3. Verificate che sia AWS ParallelCluster installato correttamente.

```
$ pcluster version
{
  "version": "3.15.0"
}
```

4. Per eseguire l'upgrade alla versione più recente, esegui nuovamente il comando di installazione.

```
$ python3 -m pip install "aws-parallelcluster" --upgrade --user
```

Installa AWS ParallelCluster come applicazione autonoma

Installa AWS ParallelCluster come applicazione autonoma nel tuo ambiente. Segui le istruzioni per l'installazione AWS ParallelCluster su un sistema operativo disponibile nella sezione seguente.

Prerequisiti

- Un ambiente con un sistema operativo compatibile con una versione disponibile del programma di installazione.

Note

AWS ParallelCluster richiede NodeJS. AWS ParallelCluster Il programma di installazione include una versione in bundle di NodeJS (v18), che viene installata se non esiste già. Se il sistema non è compatibile con NodeJS v18, è necessario installare NodeJS prima dell'installazione. AWS ParallelCluster

Linux

Linux x86 (64-bit)

Installa AWS ParallelCluster nel tuo ambiente.

1. Scarica l'ultimo programma di [installazione di pcluster](#) ([checksum](#)).
2. Decomprimi il pacchetto di installazione e installalo AWS ParallelCluster usando i seguenti comandi:

```
$ unzip pcluster-installer-bundle-<VERSION>-Linux_x86_64-signed.zip -d pcluster-  
installer-bundle  
$ cd pcluster-installer-bundle  
$ chmod +x install_pcluster.sh
```

3. Esegui il seguente script di installazione.

```
$ bash install_pcluster.sh
```

4. Verifica che AWS ParallelCluster sia installato correttamente.

```
$ pcluster version  
{  
  "version": "3.15.0"  
}
```

Risoluzione degli errori di installazione di pcluster

- Se la AWS ParallelCluster versione non viene restituita nel passaggio 4, riavvia il terminale o `source bash_profile` aggiorna la PATH variabile per includere la nuova directory binaria, come mostrato nell'esempio seguente:

```
$ source ~/.bash_profile
```

- Se utilizzi l'`pcluster` installazione per creare cluster con risorse HTTPS CustomActions specificate come HTTPS, anziché URI S3, potresti visualizzare un WARNING messaggio che indica che queste risorse potrebbero non essere verificate ([SSL : CERTIFICATE_VERIFY_FAILED]). Ciò è causato da un problema noto e puoi ignorare questo avviso se ti fidi dell'autenticità delle risorse specificate.

Versioni precedenti del pacchetto di installazione

- Nessuno

Passaggi da eseguire dopo l'installazione

Questa sezione presuppone che tu abbia installato AWS ParallelCluster. Imparerai come verificare che sia AWS ParallelCluster installato correttamente, come eseguire l'aggiornamento alla versione più recente di AWS ParallelCluster e come disinstallarlo.

Puoi verificare che sia AWS ParallelCluster stato installato correttamente eseguendo [pcluster version](#).

```
$ pcluster version
{
  "version": "3.15.0"
}
```

AWS ParallelCluster viene aggiornato regolarmente. Per eseguire l'aggiornamento alla versione più recente di AWS ParallelCluster, eseguire nuovamente il comando di installazione. Per ulteriori informazioni sulla versione più recente di AWS ParallelCluster, consulta le [note di AWS ParallelCluster rilascio](#).

```
$ pip3 install aws-parallelcluster --upgrade --user
```

Per disinstallarlo AWS ParallelCluster, usa `pip3 uninstall`.

```
$ pip3 uninstall aws-parallelcluster
```

Se non disponi di Python e `pip3`, usa la procedura per l'ambiente in uso.

Installazione del PCUI

L'AWS ParallelCluster interfaccia utente (PCUI) è un'interfaccia utente basata sul Web che rispecchia la AWS ParallelCluster `pccluster` CLI e offre un'esperienza simile a quella di una console. Puoi installare e accedere alla PCUI nel tuo Account AWS. Quando lo esegui, il PCUI accede a un'istanza dell'AWS ParallelCluster API ospitata su Amazon API Gateway nel tuo Account AWS. Per ulteriori informazioni sul PCUI, consulta [AWS ParallelCluster UI](#).

Prerequisiti:

- È necessario disporre di un Account AWS
- Devi avere accesso a Console di gestione AWS

Per ulteriori informazioni, consulta [Configurare un Account AWS](#).

Argomenti

- [Installa la PCUI](#)
- [Parametri dello stack](#)
- [Configura un dominio personalizzato](#)
- [Opzioni del pool di utenti di Amazon Cognito](#)
- [Identifica la versione e la versione PCUI AWS ParallelCluster](#)
- [Costi PCUI](#)

Installa la PCUI

Per installare un'istanza dell'AWS ParallelCluster interfaccia utente (PCUI), scegli un link di CloudFormation creazione rapida Regione AWS in cui creare i cluster. L'URL di creazione rapida ti porta a una procedura guidata di creazione dello stack in cui fornisci input per la creazione rapida del modello di stack e distribuisce lo stack. [Per ulteriori informazioni sulla creazione rapida di stack](#),

[consulta Creazione CloudFormation rapida di link per la creazione rapida di stack nella Guida per l'utente.AWS CloudFormation](#)

 Note

È possibile creare e modificare cluster o creare immagini solo con la stessa AWS ParallelCluster versione utilizzata per installare il PCUI.

Usa un link di AWS CloudFormation creazione rapida per distribuire uno stack PCUI con stack Amazon Cognito, API Gateway e Amazon EC2 Systems Manager annidati.

1. Accedi alla Console di gestione AWS.
2. Implementa il PCUI scegliendo un link per la creazione rapida dall'elenco qui. Regione AWS Questo ti porta alla procedura guidata CloudFormation Create Stack nella console.

- [us-east-1](#)
- [us-east-2](#)
- [us-west-1](#)
- [us-west-2](#)
- [eu-west-1](#)
- [eu-west-2](#)
- [eu-west-3](#)
- [eu-central-1](#)
- [eu-north-1](#)
- [me-south-1](#)
- [sa-east-1](#)
- [ca-central-1](#)
- [ap-northeast-1](#)
- [ap-northeast-2](#)
- [ap-south-1](#)
- [ap-southeast-1](#)
- [ap-southeast-2](#)
- [us-gov-west-1](#)

3. Inserisci un indirizzo email valido per l'e-mail dell'amministratore e una ParallelCluster versione.

Una volta completata correttamente la distribuzione, il PCUI invierà una password temporanea a questo indirizzo e-mail che potrai utilizzare per accedere al PCUI. Se si elimina l'e-mail prima di salvare o utilizzare la password temporanea, è necessario eliminare lo stack e reinstallare il PCUI.

4. Mantieni il resto del modulo vuoto o inserisci i valori per i parametri (facoltativi) per personalizzare la build PCUI.
5. Prendi nota del nome dello stack da utilizzare nei passaggi successivi.
6. Passa a Funzionalità. Accetta le CloudFormation funzionalità.
7. Scegli Create (Crea). Sono necessari circa 15 minuti per completare l'implementazione di AWS ParallelCluster API e PCUI.
8. Visualizza i dettagli dello stack man mano che lo stack viene creato.
9. Una volta completata la distribuzione, apri l'e-mail dell'amministratore che è stata inviata all'indirizzo che hai inserito e che contiene la password temporanea. Usalo per accedere al PCUI. (Ricorda che se elimini definitivamente l'e-mail prima di accedere al PCUI, devi eliminare lo stack PCUI che hai creato e reinstallare il PCUI.
10. Nell'elenco degli stack della CloudFormation console, scegli il link al nome dello stack che hai annotato nel passaggio precedente.
11. Nei dettagli dello stack, scegli Outputs e seleziona il link per la chiave denominata **Stackname**URL per aprire il PCUI (dove **Stackname** è il nome che hai annotato nel passaggio precedente).
12. Inserisci la password temporanea. Segui i passaggi per creare la tua password e accedere nuovamente.
13. Ora ti trovi nella home page del PCUI nella pagina Regione AWS che hai selezionato.
14. Per iniziare a usare il PCUI, vedi. [Configurazione e creazione di un cluster con PCUI](#)

Note

Le sessioni PCUI hanno una durata predefinita di 5 minuti, che è il valore minimo fornito da Cognito a partire da PCUI 2023.12.0. Pertanto, si prevede che un utente rimosso dai pool di utenti di Cognito sarà ancora in grado di accedere al sistema fino alla scadenza della sessione.

Parametri dello stack

AdditionalPoliciesPCAPI:

Descrizione: (Facoltativo) ARN della policy IAM aggiuntiva da allegare al ruolo di esecuzione predefinito per la funzione Lambda ParallelCluster . È possibile specificare una sola politica.

Tipo: String

Impostazione predefinita:»

AllowedPattern: «^ (arn:. *:sono:: *:politica\\V([a-zA-Z0-9_-] +)) | () \$»

AdminUserEmail:

Descrizione: indirizzo e-mail dell'utente amministrativo da configurare per impostazione predefinita (solo con nuove istanze di Cognito).

Tipo: String

Predefinito:»

CognitoCustomDomain:

Descrizione: (Facoltativo) Nome di dominio personalizzato per Cognito. Se omissso, verrà utilizzato il nome di dominio Cognito predefinito.

Tipo: String

Predefinito:»

CognitoCustomDomainCertificateArn:

Descrizione: (Facoltativo) ARN del certificato ACM emesso per il dominio personalizzato Cognito. Questo è obbligatorio solo se CognitoCustomDomain specificato.

Tipo: String

Impostazione predefinita:»

CustomDomain:

Descrizione: (Facoltativo) Nome di dominio personalizzato. Se omissso, verrà utilizzato il nome di dominio predefinito.

Tipo: String

Predefinito:»

CustomDomainCertificateArn:

Descrizione: (Facoltativo) ARN del certificato ACM emesso per il dominio personalizzato. È richiesto solo se specificatoCustomDomain.

Tipo: String

Impostazione predefinita:»

IAMRoleAndPolicyPrefix:

Descrizione: prefisso applicato al nome di ogni ruolo e policy IAM (lunghezza massima: 10). [ParallelCluster >= 3.8.0]

Tipo: String

Predefinito:»

MaxLength: 10

ImageBuilderSubnetId:

Descrizione: (Facoltativo) Seleziona la sottorete da utilizzare per creare le immagini del contenitore. La sottorete deve essere pubblica e deve essere assegnata automaticamente al pubblico. IPs Se non è selezionata, verrà utilizzata la sottorete predefinita.

Tipo: String

Predefinito:»

ImageBuilderVpcId:

Descrizione: (Facoltativo) Seleziona il VPC da utilizzare per creare le immagini del contenitore. Se non è selezionato, verrà utilizzato il VPC predefinito.

Tipo: String

Impostazione predefinita:»

InfrastructureBucket:

Descrizione: (Opzionale) bucket S3 in cui sono archiviati CloudFormation i file. Modificate questo parametro solo durante il test delle modifiche apportate all'infrastruttura stessa.

Tipo: String

Impostazione predefinita:»

LambdaSecurityGroupIds:

Descrizione: elenco separato da virgole dei gruppi di sicurezza da associare alla funzione PCUI Lambda.

Tipo: CommaDelimitedList

Impostazione predefinita:»

LambdaSubnetIds:

Descrizione: elenco separato da virgole di sottorete IDs da associare alla funzione PCUI Lambda. Queste sottoreti devono essere private e associate all'endpoint VPC.

Tipo: CommaDelimitedList

Predefinito:»

PermissionsBoundaryPolicy:

Descrizione: ARN della policy IAM da utilizzare come limite delle autorizzazioni per ogni ruolo IAM creato dall' ParallelCluster infrastruttura dell'interfaccia utente. '

Tipo: String

Predefinito:»

AllowedPattern: «^ (arn:. *:sono::. *:politica\\([a-zA-Z0-9_-] +)) | () \$»

PermissionsBoundaryPolicyPCAPI:

Descrizione: ARN della policy IAM da utilizzare come limite di autorizzazioni per ogni ruolo IAM creato dall'infrastruttura API. ParallelCluster [>= 3.8.0] ParallelCluster

Tipo: String

Predefinito:»

AllowedPattern: «^ (arn:. *:sono::. *:politica\\([a-zA-Z0-9_-] +)) | () \$»

PublicEcrImageUri:

Descrizione: quando specificato, l'URI dell'immagine Docker per la Lambda del contenitore ParallelCluster dell'interfaccia utente.

Tipo: String

Impostazione predefinita: public.ecr.aws/pcm/parallelcluster-ui:2024.11.0

SNSRole:

Descrizione: SNSRole ARN di uno stack PCUI Cognito Stack precedentemente distribuito. Lascia vuoto per crearne uno nuovo.

Tipo: String

Predefinito:»

UserPoolAuthDomain:

Descrizione: UserPoolAuthDomain di un pool di utenti PCUI Cognito distribuito in precedenza. Lascia vuoto per crearne uno nuovo.

Tipo: String

Predefinito:»

UserPoolId:

Descrizione: UserPoolId di un pool di utenti PCUI Cognito distribuito in precedenza. Lascia vuoto per crearne uno nuovo.

Tipo: String

Predefinito:»

Versione:

Descrizione: versione di AWS ParallelCluster da distribuire.

Tipo: String

Predefinito: 3.11.1

VpcEndpointId:

Descrizione: immettere un endpoint VPC con interfaccia di tipo per il servizio execute-api per abilitare l'implementazione PCUI privata. Se abilitata, l'API accetterà solo richieste dall'interno del VPC specificato.

Tipo: String

Impostazione predefinita:»

Configura un dominio personalizzato

Scopri come configurare un dominio personalizzato per l' AWS ParallelCluster interfaccia utente (PCUI). L'interfaccia utente è ospitata su Amazon API Gateway nel tuo Account AWS. Puoi configurare un dominio personalizzato nella console API Gateway.

Prerequisiti:

- Hai un Account AWS.
- Possiedi un dominio.
- La radice del dominio è risolvibile. Ad esempio, se si desidera utilizzare `xyz.example.com` come dominio personalizzato, il dominio principale `example.com` deve essere risolvibile. Ciò significa che è necessario disporre almeno di un record A per `example.com` puntare a un IP. Se non disponi di tale record, aggiungi un record fittizio che punti a 0.0.0.0.
- Hai creato o importato il certificato di dominio personalizzato per Cognito in AWS Certificate Manager (ACM) in us-east-1.
- Disponi delle autorizzazioni necessarie per modificare le impostazioni del Domain Name System (DNS).

Argomenti

- [Distribuisci PCUI](#)
- [Configurazione di DNS](#)

Distribuisci PCUI

Completa i seguenti passaggi per creare o aggiornare la distribuzione AWS ParallelCluster dell'interfaccia utente (PCUI) in modo che utilizzi il tuo dominio personalizzato.

In questo esempio supponiamo che tu voglia distribuire PCUI con un dominio personalizzato `xyz.example.com` e l'interfaccia Amazon Cognito con un dominio personalizzato. `auth-xyz.example.com`

Note

Tieni presente che la personalizzazione del dominio Amazon Cognito non è obbligatoria e può essere lasciata con il valore predefinito. Includiamo questa personalizzazione per completezza.

Note

I domini personalizzati per Amazon Cognito non sono supportati in. AWS GovCloud (US) Regions

A tale scopo, distribuisce lo stack PCUI con i seguenti parametri:

- CustomDomain: *xyz.example.com*.
- CustomDomainCertificateArn: il certificato ACM Amazon Resource Name (ARN) per. *xyz.example.com*
- CognitoCustomDomain: *auth-xyz.example.com*.
- CognitoCustomDomainCertificateArn: il certificato ACM Amazon Resource Name (ARN) per. *xyz.example.com*

Configurazione di DNS

Completa i seguenti passaggi per configurare il tuo DNS (Domain Name Service) in modo che l'AWS ParallelCluster interfaccia utente (PCUI) e Amazon Cognito rispondano ai domini personalizzati desiderati.

Questo esempio presuppone che tu voglia distribuire PCUI con dominio personalizzato `xyz.example.com` e l'interfaccia Amazon Cognito con dominio personalizzato. `auth-xyz.example.com`

1. Distribuisce lo stack PCUI con i seguenti parametri:

- CustomDomainEndpoint: crea un record A nel tuo DNS *xyz.example.com* perché punti all'alias specificato nell'output.
 - CognitoCustomDomainEndpoint: Crea un record A nel tuo DNS *auth-xyz.example.com* perché punti all'alias specificato nell'output.
2. Attendi circa 10 minuti in modo che le modifiche al DNS possano essere distribuite.
 3. Quando le modifiche DNS vengono distribuite, PCUI risponde *xyz.example.com/pcui* e la pagina di autenticazione di Amazon Cognito risponde. *auth-xyz.example.com*

Opzioni del pool di utenti di Amazon Cognito

Le sezioni seguenti fanno riferimento ai collegamenti di creazione CloudFormation rapida o alla creazione rapida. URLs Un URL di creazione rapida consente di accedere a una procedura guidata di creazione dello stack, in cui è possibile fornire input rapidi per il modello di stack e distribuire lo stack. [Per ulteriori informazioni sulla creazione rapida di stack, consulta Creazione CloudFormation rapida di link per la creazione rapida di stack nella Guida per l'utente.AWS CloudFormation](#)

Per mantenere un pool di utenti Amazon Cognito da utilizzare con più istanze AWS ParallelCluster UI (PCUI), considera le seguenti opzioni:

- Usa un'istanza PCUI esistente che si collega a un pool di utenti Amazon Cognito creato da uno CloudFormation stack annidato. Questo è ciò che viene creato quando distribuisce il PCUI utilizzando il link di creazione rapida e mantieni vuoti tutti i parametri di Amazon Cognito.
- Innanzitutto, distribuisce un pool di utenti Amazon Cognito autonomo. Quindi, distribuisce una nuova istanza PCUI collegata al pool di utenti autonomo di Amazon Cognito che hai appena distribuito. In questo modo, separi la distribuzione di Amazon Cognito dalla distribuzione PCUI. Tieni presente che gli CloudFormation stack PCUI non annidati sono più facili da aggiornare.

Usa un pool di utenti Amazon Cognito esistente con una nuova istanza PCUI

Completa i seguenti passaggi per utilizzare un pool di utenti Amazon Cognito esistente con una nuova istanza PCUI.

Usa un pool di utenti Amazon Cognito esistente con una nuova istanza PCUI

1. Nella CloudFormation console, seleziona lo stack PCUI che contiene il pool di utenti di Amazon Cognito che desideri utilizzare con più istanze PCUI.

2. Passa allo stack annidato che ha creato il pool di utenti Amazon Cognito.
3. Selezionare la scheda Outputs (Output).
4. Copia i valori dei seguenti parametri:
 - UserPoolId
 - UserPoolAuthDomain
 - SNSRole
5. Distribuisci una nuova istanza PCUI utilizzando il link di creazione rapida e compila tutti i External PCUI Amazon Cognito parametri con gli output che hai copiato. Ciò impedisce al nuovo stack PCUI di creare un nuovo pool e lo collega al pool di utenti Amazon Cognito esistente creato da uno stack annidato. Puoi distribuire nuove istanze PCUI successive con gli stessi valori dei parametri e collegarle al pool di utenti di Amazon Cognito.

Crea un pool di utenti Amazon Cognito autonomo

Per creare un pool di utenti Amazon Cognito autonomo

1. Avvia uno stack solo per Amazon Cognito scegliendo un link di creazione rapida etichettato con lo stesso Regione AWS in cui distribuisce le tue istanze PCUI da questo elenco:
 - [us-east-1](#)
 - [us-east-2](#)
 - [us-west-1](#)
 - [us-west-2](#)
 - [eu-west-1](#)
 - [eu-west-2](#)
 - [eu-west-3](#)
 - [eu-central-1](#)
 - [eu-north-1](#)
 - [me-south-1](#)
 - [sa-east-1](#)
 - [ca-central-1](#)
 - [ap-northeast-1](#)
 - [ap-northeast-2](#)

- [ap-south-1](#)
 - [ap-southeast-1](#)
 - [ap-southeast-2](#)
 - [us-gov-west-1](#)
2. Al termine della creazione dello stack, selezionate la scheda Output e copiate i valori dei seguenti parametri:
 - UserPoolId
 - UserPoolAuthDomain
 - SNSRole
 3. Distribuisci una nuova istanza PCUI scegliendo un link di avvio rapido PCUI e compilando tutti i External PCUI Amazon Cognito parametri con i valori copiati. La nuova istanza PCUI si collega al pool di utenti autonomo di Amazon Cognito e non crea uno stack annidato o un nuovo pool di utenti. Puoi distribuire nuove istanze PCUI successive con gli stessi valori dei parametri e collegarle al pool di utenti autonomo di Amazon Cognito.

Identifica la versione e la versione PCUI AWS ParallelCluster

Per identificare la versione PCUI AWS ParallelCluster e PCUI:

1. Nella CloudFormation console, seleziona uno stack PCUI.
2. Seleziona la scheda Parametri.
3. La AWS ParallelCluster versione è il valore del parametro Version.
4. La versione PCUI si trova alla fine del PublicEcrImageUri valore. Ad esempio, se il valore è `public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02`, allora la versione è `2023.02`.

Note

Per aggiornare PCUI alla AWS ParallelCluster versione più recente, avvia un nuovo stack scegliendo un link di creazione [rapida](#).

Costi PCUI

Il PCUI è basato su un'architettura serverless ed è possibile utilizzarlo nella categoria AWS Free Tier nella maggior parte dei casi. Nella tabella seguente sono elencati i fattori da cui dipende il PCUI e i relativi limiti relativi al livello gratuito. Servizi AWS Si stima che il costo di utilizzo tipico sia inferiore a un dollaro al mese.

Servizio	AWS Livello gratuito
Amazon Cognito	50.000 utenti attivi al mese
Gateway Amazon API	1 milione di chiamate API REST
AWS Lambda	1 milione di richieste gratuite ogni mese e 400.000 GB di tempo di elaborazione ogni mese
EC2 Image Builder	Nessun costo, tranne EC2
Amazon Elastic Compute Cloud	Creazione unica dell'immagine del contenitore in 15 minuti
CloudFormation	5 GB di dati (acquisizione, archiviazione e dati scansionati mediante query di Logs Insights)

Nozioni di base su AWS ParallelCluster

Per iniziare a AWS ParallelCluster configurare e creare un cluster utilizzando l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'interfaccia utente (UI) basata sul Web. Il PCUI è stato aggiunto nella versione 3.5.0.

Argomenti

- [Configura e crea un cluster con AWS ParallelCluster interfaccia a riga di comando](#)
- [Configurazione e creazione di un cluster con PCUI](#)
- [Connessione a un cluster](#)

Configura e crea un cluster con AWS ParallelCluster interfaccia a riga di comando

Dopo l'installazione AWS ParallelCluster, completa i seguenti passaggi di configurazione.

1. Verifica che il tuo AWS account abbia un ruolo che includa le autorizzazioni necessarie per eseguire la `pcluster` CLI. Per ulteriori informazioni, consulta [AWS ParallelCluster esempi di politiche pcluster utente](#).
2. Configura le tue credenziali. AWS Per ulteriori informazioni, consulta [Configurazione della AWS CLI](#) nella Guida per l'utente di AWS CLI .

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [us-east-1]: us-east-1
Default output format [None]:
```

3. Il Regione AWS luogo in cui viene lanciato il cluster deve avere almeno una coppia di chiavi Amazon EC2. Per ulteriori informazioni, consulta le [coppie di chiavi di Amazon Elastic Compute Cloud](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI), si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Configura e crea il tuo primo cluster

Per creare il primo cluster, utilizza il comando `pcluster configure` CLI per avviare una procedura guidata che richiede tutte le informazioni necessarie per configurare e creare il cluster. I dettagli della sequenza differiscono quando viene utilizzata AWS Batch come scheduler rispetto all'utilizzo. Slurm

Slurm

```
$ pcluster configure --config cluster-config.yaml
```

Dall'elenco degli Regione AWS identificatori validi, scegli Regione AWS dove vuoi che venga eseguito il cluster.

 Note

L'elenco Regioni AWS visualizzato si basa sulla partizione del tuo account e include Regioni AWS solo quelle abilitate per il tuo account. Per ulteriori informazioni sull'attivazione Regioni AWS del tuo account, consulta [Gestione Regioni AWS](#) in. Riferimenti generali di AWS L'esempio mostrato proviene dalla partizione AWS Global. Se il tuo account è nella AWS GovCloud (US) partizione, solo Regioni AWS in quella partizione sono elencati (gov-us-east-1). gov-us-west-1 Allo stesso modo, se il tuo account si trova nella partizione AWS Cina, cn-northwest-1 vengono visualizzati solo i cn-north-1 e. Per l'elenco completo dei servizi Regioni AWS supportati da AWS ParallelCluster, vedi [Supportato Regioni AWS per AWS ParallelCluster](#).

Allowed values for Regione AWS ID:

1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ap-southeast-3
9. ap-southeast-5
10. ap-southeast-7
11. ca-central-1
12. eu-central-1
13. eu-north-1
14. eu-south-1
15. eu-west-1
16. eu-west-2
17. eu-west-3
18. il-central-1
19. me-south-1
20. sa-east-1
21. us-east-1
22. us-east-2
23. us-west-1
24. us-west-2

Regione AWS ID [ap-northeast-1]:

Scegli la coppia di key pair tra quelle registrate con Amazon Elastic Compute Cloud tra quelle selezionate Regione AWS:

Allowed values for Amazon EC2 Key Pair Name:

1. your-key-1
2. your-key-2

Amazon EC2 Key Pair Name [your-key-1]:

Scegliere il pianificatore da utilizzare con il cluster.

Allowed values for Scheduler:

1. slurm
2. awsbatch

Scheduler [slurm]:

Scegliere il sistema operativo.

Allowed values for Operating System:

1. alinux2
2. alinux2023
3. ubuntu2404
4. ubuntu2204
6. rhel8
7. rhel9

Operating System [alinux2]:

Scegli il tipo di istanza del nodo principale:

Head node instance type [t2.micro]:

Scegli la configurazione della coda. Nota: il tipo di istanza non può essere specificato per più risorse di calcolo nella stessa coda.

Number of queues [1]:

Name of queue 1 [queue1]:

Number of compute resources for queue1 [1]: 2

Compute instance type for compute resource 1 in queue1 [t2.micro]:

Maximum instance count [10]:

Consenti a EFA di eseguire applicazioni che richiedono alti livelli di comunicazione tra istanze su AWS larga scala senza costi aggiuntivi:

- Scegli un tipo di istanza che [supporti Elastic Fabric Adapter \(EFA\)](#).
- Abilita [EFA](#).
- Specificate il nome di un [gruppo di collocamento](#) esistente. Se lo lasciate vuoto, ne AWS ParallelCluster crea uno per voi.

```
Compute instance type for compute resource 2 in queue1 [t2.micro]: c5n.18xlarge
Enable EFA on c5n.18xlarge (y/n) [y]: y
Maximum instance count [10]:
Placement Group name []:
```

Una volta completati i passaggi precedenti, decidi se utilizzare un VPC esistente o lasciare che venga AWS ParallelCluster creato un VPC per te. Se non disponi di un VPC configurato correttamente, AWS ParallelCluster puoi crearne uno nuovo per te. Posiziona sia il nodo principale che il nodo di calcolo nella stessa sottorete pubblica oppure solo il nodo principale in una sottorete pubblica con tutti i nodi di elaborazione in una sottorete privata. Se consenti di AWS ParallelCluster creare un VPC, devi decidere se tutti i nodi devono trovarsi in una sottorete pubblica. Per ulteriori informazioni, consulta [Configurazioni di rete](#).

Se configuri il cluster per utilizzare tipi di istanze con più interfacce di rete o una scheda di rete, consulta la sezione [Configurazioni di rete](#) per ulteriori requisiti di rete.

È possibile raggiungere la quota stabilita per il numero di VPC consentiti in a. Regione AWS La quota predefinita è di cinque VPC per un. Regione AWS Per ulteriori informazioni su questa quota e su come richiedere un aumento, consulta [VPC e sottoreti nella Amazon VPC User Guide](#).

Important

I VPC creati da AWS ParallelCluster non abilitano i log di flusso VPC per impostazione predefinita. I log di flusso VPC ti consentono di acquisire informazioni sul traffico IP in entrata e in uscita dalle interfacce di rete nei tuoi VPC. Per ulteriori informazioni, consulta [Log di flusso VPC](#) nella Guida per l'utente di Amazon VPC.

Se consenti di AWS ParallelCluster creare un VPC, assicurati di decidere se tutti i nodi devono trovarsi in una sottorete pubblica.

Note

Se lo desideri 1. Head node in a public subnet and compute fleet in a private subnet, AWS ParallelCluster crea un gateway NAT che comporta costi aggiuntivi, anche se specifichi risorse di livello gratuito.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Se non crei un nuovo VPC, devi selezionare un VPC esistente.

Se scegli di AWS ParallelCluster creare il VPC, prendi nota dell'ID VPC in modo da poterlo utilizzare AWS CLI per eliminarlo in un secondo momento.

```
Automate VPC creation? (y/n) [n]: n
Allowed values for VPC ID:
#  id                                     name                                     number_of_subnets
---  ---
1  vpc-0b4ad9c4678d3c7ad  ParallelClusterVPC-20200118031893      2
2  vpc-0e87c753286f37eef  ParallelClusterVPC-20191118233938      5
VPC ID [vpc-0b4ad9c4678d3c7ad]: 1
```

Dopo aver selezionato il VPC, decidere se utilizzare le sottoreti esistenti o crearne di nuove.

```
Automate Subnet creation? (y/n) [y]: y
```

```
Creating CloudFormation stack...  
Do not leave the terminal until the process has finished
```

AWS Batch

```
$ pcluster configure --config cluster-config.yaml
```

Dall'elenco degli Regione AWS identificatori validi, scegli Regione AWS dove vuoi che venga eseguito il cluster.

Note

L'elenco Regioni AWS visualizzato si basa sulla partizione del tuo account. Include solo Regioni AWS quelle abilitate per il tuo account. Per ulteriori informazioni sull'attivazione Regioni AWS del tuo account, consulta [Gestione Regioni AWS](#) in Riferimenti generali di AWS. L'esempio mostrato proviene dalla partizione AWS Global. Se il tuo account è nella AWS GovCloud (US) partizione, solo Regioni AWS in quella partizione sono elencati (gov-us-east-1e). gov-us-west-1 Allo stesso modo, se il tuo account si trova nella partizione AWS Cina, cn-northwest-1 vengono visualizzati solo i cn-north-1 e. Per l'elenco completo dei servizi Regioni AWS supportati da AWS ParallelCluster, vedi [Supportato Regioni AWS per AWS ParallelCluster](#).

Allowed values for Regione AWS ID:

1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ap-southeast-3
9. ap-southeast-5
10. ap-southeast-7
11. ca-central-1
12. eu-central-1
13. eu-north-1
14. eu-south-1
15. eu-west-1
16. eu-west-2


```
17. eu-west-3
18. il-central-1
19. me-south-1
20. sa-east-1
21. us-east-1
22. us-east-2
23. us-west-1
24. us-west-2
Regione AWS ID [us-east-1]:
```

La coppia di chiavi viene selezionata tra le coppie di chiavi registrate con Amazon EC2 nel gruppo selezionato. Regione AWS Scegli la key pair:

```
Allowed values for Amazon EC2 Key Pair Name:
1. your-key-1
2. your-key-2
Amazon EC2 Key Pair Name [your-key-1]:
```

Scegliere il pianificatore da utilizzare con il cluster.

```
Allowed values for Scheduler:
1. slurm
2. awsbatch
Scheduler [slurm]: 2
```

Quando come pianificatore è selezionato `awsbatch`, come sistema operativo viene utilizzato `alinux2`. Viene inserito il tipo di istanza del nodo principale:

```
Head node instance type [t2.micro]:
```

Scegliete la configurazione della coda. Lo AWS Batch scheduler contiene solo una singola coda. Viene inserita la dimensione massima del cluster di nodi di elaborazione. Questa viene misurata in vCPU.

```
Number of queues [1]:
Name of queue 1 [queue1]:
Maximum vCPU [10]:
```

Decidi se utilizzare i VPC esistenti o lasciarti AWS ParallelCluster creare i VPC per te. Se non si dispone di un VPC configurato correttamente, è possibile crearne uno nuovo con AWS

ParallelCluster . Utilizza sia il nodo principale che il nodo di calcolo nella stessa sottorete pubblica o solo il nodo principale in una sottorete pubblica con tutti i nodi in una sottorete privata. È possibile raggiungere la propria quota sul numero di VPC consentiti in una regione. Il numero predefinito di VPC è cinque. Per ulteriori informazioni su questa quota e su come richiedere un aumento, consulta [VPC e sottoreti nella Amazon VPC User Guide](#).

 Important

I VPC creati da AWS ParallelCluster non abilitano i log di flusso VPC per impostazione predefinita. I log di flusso VPC ti consentono di acquisire informazioni sul traffico IP in entrata e in uscita dalle interfacce di rete nei tuoi VPC. Per ulteriori informazioni, consulta [Log di flusso VPC](#) nella Guida per l'utente di Amazon VPC.

Se consenti di AWS ParallelCluster creare un VPC, assicurati di decidere se tutti i nodi devono trovarsi in una sottorete pubblica.

 Note

Se lo desideri 1. Head node in a public subnet and compute fleet in a private subnet, AWS ParallelCluster crea un gateway NAT che comporta costi aggiuntivi, anche se specifichi risorse di livello gratuito.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
subnet]: *1*
```

```
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Se non crei un nuovo VPC, devi selezionare un VPC esistente.

Se scegli di AWS ParallelCluster creare il VPC, prendi nota dell'ID VPC in modo da poterlo utilizzare AWS CLI o Console di gestione AWS per eliminarlo in un secondo momento.

```
Automate VPC creation? (y/n) [n]: n
Allowed values for VPC ID:
#   id                               name                               number_of_subnets
---  -----
  1  vpc-0b4ad9c4678d3c7ad  ParallelClusterVPC-20200118031893          2
  2  vpc-0e87c753286f37eef  ParallelClusterVPC-20191118233938          5
VPC ID [vpc-0b4ad9c4678d3c7ad]: 1
```

Dopo aver selezionato il VPC, assicurati di decidere se utilizzare le sottoreti esistenti o crearne di nuove.

```
Automate Subnet creation? (y/n) [y]: y
```

```
Creating CloudFormation stack...
Do not leave the terminal until the process has finished
```

Una volta completati i passaggi precedenti, un semplice cluster viene avviato in un VPC. Il VPC utilizza una sottorete esistente che supporta gli indirizzi IP pubblici. La tabella di routing per la sottorete è. `0.0.0.0/0 => igw-xxxxxx` Nota le seguenti condizioni:

- Il VPC deve avere DNS Resolution = yes e DNS Hostnames = yes.
- Il VPC deve inoltre disporre di opzioni DHCP con la corretta domain-name per. Regione AWS Il set opzioni DHCP predefinito specifica già il AmazonProvidedDNS necessario. Se specifichi più di un server di nomi di dominio, consulta i [set di opzioni DHCP](#) nella Amazon VPC User Guide. Quando utilizzi sottoreti private, utilizza un gateway NAT o un proxy interno per abilitare l'accesso web per i nodi di calcolo. Per ulteriori informazioni, consulta [Configurazioni di rete](#).

Quando tutte impostazioni contengono valori validi, è possibile avviare il cluster eseguendo il comando creato:

```
$ pcluster create-cluster --cluster-name test-cluster --cluster-configuration cluster-  
config.yaml  
{  
  "cluster": {  
    "clusterName": "test-cluster",  
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",  
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/test-cluster/  
abcdef0-f678-890a-5abc-021345abcdef",  
    "region": "eu-west-1",  
    "version": "3.15.0",  
    "clusterStatus": "CREATE_IN_PROGRESS"  
  },  
  "validationMessages": []  
}
```

Segui l'avanzamento del cluster:

```
$ pcluster describe-cluster --cluster-name test-cluster
```

or

```
$ pcluster list-clusters --query 'clusters[?clusterName==`test-cluster`]'
```

Dopo che il cluster ha raggiunto "clusterStatus": "CREATE_COMPLETE" lo stato, puoi connetterti ad esso utilizzando le normali impostazioni del client SSH. Per ulteriori informazioni sulla connessione alle istanze Amazon EC2, consulta la Amazon EC2 User Guide [nella Amazon EC2 User Guide](#). Oppure puoi connettere il cluster tramite

```
$ pcluster ssh --cluster-name test-cluster -i ~/path/to/keyfile.pem
```

Per eliminare il cluster, esegui il comando seguente.

```
$ pcluster delete-cluster --region us-east-1 --cluster-name test-cluster
```

Dopo l'eliminazione del cluster, puoi eliminare le risorse di rete nel VPC eliminando lo stack di rete. CloudFormation Il nome dello stack inizia con «parallelclusternetworking-» e contiene l'ora di creazione nel formato «YYYYMMDDHHMMSS». È possibile [list-stacks](#) elencare gli stack utilizzando il comando.

```
$ aws --region us-east-1 cloudformation list-stacks \
  --stack-status-filter "CREATE_COMPLETE" \
  --query "StackSummaries[].StackName" | \
  grep -e "parallelclusternetworking-"
"parallelclusternetworking-pubpriv-20191029205804"
```

Lo stack può essere eliminato utilizzando il [delete-stack](#) comando.

```
$ aws --region us-east-1 cloudformation delete-stack \
  --stack-name parallelclusternetworking-pubpriv-20191029205804
```

Il VPC che [pcluster configure](#) crea per te non viene creato nello stack di CloudFormation rete. È possibile eliminare il VPC manualmente nella console o utilizzando il AWS CLI

```
$ aws --region us-east-1 Amazon EC2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

Configurazione e creazione di un cluster con PCUI

La PCUI è un'interfaccia utente basata sul Web che rispecchia la AWS ParallelCluster `pcluster` CLI, fornendo al contempo un'esperienza simile a quella di una console. Si installa e si accede al PCUI dal proprio Account AWS. Quando lo esegui, il PCUI accede a un'istanza dell' AWS ParallelCluster API ospitata su Amazon API Gateway nel tuo Account AWS.

Note

La procedura guidata PCUI potrebbe non disporre di opzioni di interfaccia utente per tutte le funzionalità supportate nell'ultima versione supportata. AWS ParallelCluster. È possibile modificare manualmente il file di configurazione in base alle esigenze o utilizzare la AWS ParallelCluster CLI.

Questa sezione illustra come configurare e creare un cluster utilizzando il PCUI.

Prerequisiti:

- Accesso a un'istanza in esecuzione di PCUI. Per ulteriori informazioni, consulta [Installazione del PCUI](#).

Configura e crea un cluster

1. Nella vista Clusters PCUI, scegli Crea cluster, Passo dopo passo.
2. In Cluster, Name, inserisci un nome per il tuo cluster.
3. Scegli un VPC con una sottorete pubblica per il cluster, quindi scegli Avanti.
4. Nel nodo Head, scegli Aggiungi sessione SSM, quindi scegli Avanti.
5. In Queues, Compute resources, scegli 1 per i nodi statici.
6. Per Tipo di istanza, rimuovi il tipo di istanza predefinito selezionato, scegli t2.micro, quindi scegli Avanti.
7. In Archiviazione, scegli Avanti.
8. Nella configurazione del cluster, esamina la configurazione del cluster YAML e scegli Dry run per convalidarla.
9. Scegli Crea per creare il tuo cluster, in base alla configurazione convalidata.
10. Dopo alcuni secondi, il PCUI ti riporta automaticamente a Clusters, dove puoi monitorare lo stato di creazione del cluster e gli eventi dello stack.
11. Scegli Dettagli per visualizzare i dettagli del cluster, come la versione e lo stato.
12. Scegli Istanze per visualizzare l'elenco delle istanze Amazon EC2 e lo stato.
13. Scegli Stack events per visualizzare gli eventi dello stack del cluster e un Console di gestione AWS link allo CloudFormation stack che crea il cluster.
14. In Dettagli, una volta completata la creazione del cluster, scegli Visualizza YAML per visualizzare o scaricare il file YAML di configurazione del cluster.
15. Una volta completata la creazione del cluster, scegli Shell per accedere al nodo principale del cluster.

Note

Quando scegli Shell, AWS ParallelCluster apre una sessione di Amazon EC2 Systems Manager e ne aggiunge `ssm-user` una `/etc/sudoers` a. Per ulteriori informazioni, consulta [Attivare o disattivare le autorizzazioni amministrative ssm-user dell'account nella Guida](#) per l'utente di Amazon EC2 Systems Manager.

16. Per eseguire la pulizia, nella vista Cluster, seleziona il cluster e scegli Azioni, Elimina cluster.

Connessione a un cluster

Quando si utilizza AWS ParallelCluster, è possibile connettersi al nodo principale del cluster per eseguire lavori, visualizzare i risultati, gestire gli utenti e monitorare il cluster e lo stato dei processi. Per connetterti all'istanza del nodo principale del cluster, utilizza uno dei seguenti metodi:

- Puoi usare ssh con una [key pair](#) per accedere. Specificare la chiave privata in [HeadNode/KeyName](#) nella configurazione del cluster. Per ulteriori informazioni, consulta [Connettiti alla tua istanza Linux usando SSH](#) nella Amazon EC2 User Guide for Linux Instances.
- È possibile utilizzare il `pcluster ssh` comando CLI (Command Line Interface) per accedere. Specificare la chiave privata nella configurazione del cluster [HeadNode/KeyName](#). Per ulteriori informazioni, consulta [pcluster ssh](#).
- È possibile utilizzare una sessione SSM per connettersi al nodo principale del cluster. È necessario aggiungere la politica AmazonSSMManagedInstanceCore gestita a [HeadNode/AdditionalIamPolicies](#) nella configurazione del cluster per connettersi utilizzando una sessione SSM. Per ulteriori informazioni, consulta [SSM session manager nella Guida](#) per l'utente SSM.
- Puoi usare Amazon DCV per connetterti al nodo principale del cluster. Per ulteriori informazioni, consulta [Connect ai nodi principali e di accesso tramite Amazon DCV](#).
- Quando usi il PCUI, puoi anche usare un comando Amazon EC2 Connect fornito dall'interfaccia utente per connetterti al nodo principale del cluster.

Accesso di più utenti ai cluster

Impara a implementare e gestire l'accesso di più utenti a un singolo cluster.

In questo argomento, per AWS ParallelCluster utente si intende un utente di sistema per le istanze di calcolo. Un esempio è un esempio `ec2-user` per un'istanza Amazon EC2.

AWS ParallelCluster il supporto per l'accesso multiutente è disponibile Regioni AWS ovunque AWS ParallelCluster sia attualmente disponibile. Funziona con altri Servizi AWS, tra cui [Amazon FSx for Lustre](#) e [Amazon Elastic File System](#).

È possibile utilizzare un [AWS Directory Service for Microsoft Active Directory](#) o [Simple AD](#) per gestire l'accesso al cluster. Assicurati di verificare [Regione AWS la disponibilità](#) di questi servizi. Per configurare un cluster, specifica una [AWS ParallelCluster DirectoryService](#) configurazione. AWS

Directory Service le directory possono essere collegate a più cluster. Ciò consente una gestione centralizzata delle identità in più ambienti e un'esperienza di accesso unificata.

Se si utilizza l'accesso AWS Directory Service per AWS ParallelCluster più utenti, è possibile accedere al cluster con le credenziali utente definite nella directory. Queste credenziali sono costituite da un nome utente e una password. Dopo il primo accesso al cluster, viene generata automaticamente una chiave SSH utente. È possibile utilizzarla per accedere senza password.

È possibile creare, eliminare e modificare gli utenti o i gruppi di un cluster dopo la distribuzione del servizio di directory. Con AWS Directory Service, puoi farlo in Console di gestione AWS o utilizzando lo strumento Utenti e computer di Active Directory. Questo strumento è accessibile da qualsiasi istanza Amazon EC2 aggiunta al tuo Active Directory. Per ulteriori informazioni, consulta [Installazione degli strumenti di amministrazione Active Directory](#).

Se prevedi di utilizzarlo AWS ParallelCluster in un'unica sottorete senza accesso a Internet, consulta la sezione dedicata [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#) ai requisiti aggiuntivi.

Argomenti

- [Crea un Active Directory](#)
- [Crea un cluster con un dominio AD](#)
- [Accedere a un cluster integrato con un dominio AD](#)
- [Esecuzione di processi MPI](#)
- [Esempio di configurazioni di cluster AWS Managed Microsoft AD LDAP \(S\)](#)

Crea un Active Directory

Assicurati di creare un Active Directory (AD) prima di creare il cluster. Per informazioni su come scegliere il tipo di active directory per il cluster, consulta [Quale scegliere](#) nella Guida all'AWS Directory Service amministrazione.

Se la directory è vuota, aggiungi utenti con nomi utente e password. Per ulteriori informazioni, consulta la documentazione specifica per [AWS Directory Service for Microsoft Active Directoryo Simple AD](#).

 Note

AWS ParallelCluster richiede che ogni directory utente di Active Directory sia presente nella `/home/$user` directory.

Crea un cluster con un dominio AD

 Warning

Questa sezione introduttiva descrive come eseguire la configurazione AWS ParallelCluster con un server Managed Active Directory (AD) tramite il Lightweight Directory Access Protocol (LDAP). LDAP è un protocollo non sicuro. Per i sistemi di produzione, consigliamo vivamente l'uso di certificati TLS (LDAPS) come descritto nella [Esempio di configurazioni di cluster AWS Managed Microsoft AD LDAP \(S\)](#) sezione che segue.

Configura il cluster per l'integrazione con una directory specificando le informazioni pertinenti nella `DirectoryService` sezione del file di configurazione del cluster. Per ulteriori informazioni, consulta la sezione sulla [DirectoryService](#) configurazione.

È possibile utilizzare questo esempio per integrare il cluster con un AWS Managed Microsoft AD protocollo LDAP (Lightweight Directory Access Protocol).

Definizioni specifiche richieste per una configurazione AWS Managed Microsoft AD tramite LDAP:

- È necessario impostare il `ldap_auth_disable_tls_never_use_in_production` parametro su `True` under [DirectoryService/AdditionalSssdConfigs](#).
- È possibile specificare i nomi host dei controller o gli indirizzi IP per [DirectoryService/DomainAddr](#).
- [DirectoryService](#)/la [DomainReadOnlyUser](#) sintassi deve essere la seguente:

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Ottieni i tuoi dati AWS Managed Microsoft AD di configurazione:

```
$ aws ds describe-directories --directory-id "d-abcdef01234567890"
```

```
{
  "DirectoryDescriptions": [
    {
      "DirectoryId": "d-abcdef01234567890",
      "Name": "corp.example.com",
      "DnsIpAddrs": [
        "203.0.113.225",
        "192.0.2.254"
      ],
      "VpcSettings": {
        "VpcId": "vpc-021345abcdef6789",
        "SubnetIds": [
          "subnet-1234567890abcdef0",
          "subnet-abcdef01234567890"
        ],
        "AvailabilityZones": [
          "region-idb",
          "region-idd"
        ]
      }
    }
  ]
}
```

Configurazione del cluster per AWS Managed Microsoft AD:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
```

```
MaxCount: 10
Networking:
  SubnetIds:
    - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

Per utilizzare questa configurazione per un Simple AD, modifica il valore della **DomainReadOnlyUser** proprietà nella **DirectoryService** sezione:

```
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:SimpleAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

Considerazioni:

- Si consiglia di utilizzare LDAP su TLS/SSL (o LDAPS) anziché utilizzare solo LDAP. TLS/SSL garantisce che la connessione sia crittografata.
- Il valore della [DomainAddr](#) proprietà [DirectoryService](#)/corrisponde alle voci dell'`DnsIpAddr` elenco dell'output. `describe-directories`
- È consigliabile che il cluster utilizzi sottoreti situate nella stessa zona di disponibilità a cui [DomainAddr](#) punta [DirectoryService](#)/. Se utilizzi una [configurazione DHCP \(Dynamic Host Configuration Protocol\) personalizzata](#) consigliata per la directory VPCs e le tue sottoreti non si trovano nella zona di [DomainAddr](#) disponibilità [DirectoryService](#)/, è possibile il traffico incrociato tra le zone di disponibilità. L'uso di configurazioni DHCP personalizzate non è necessario per utilizzare la funzionalità di integrazione AD multiutente.
- Il valore della [DomainReadOnlyUser](#) proprietà [DirectoryService](#)/specifica un utente che deve essere creato nella directory. Questo utente non viene creato per impostazione predefinita. Si consiglia di non concedere a questo utente l'autorizzazione a modificare i dati della directory.

- Il valore della [PasswordSecretArn](#) proprietà [DirectoryService](#)/punta a un Gestione dei segreti AWS segreto che contiene la password dell'utente specificato per la [DomainReadOnlyUser](#) proprietà [DirectoryService](#)/. Se la password di questo utente cambia, aggiorna il valore segreto e aggiorna il cluster. Per aggiornare il cluster per il nuovo valore segreto, è necessario interrompere la flotta di calcolo con il `pcluster update-compute-fleet` comando. Se hai configurato il cluster per l'uso [LoginNodes](#), interrompi [LoginNodes/Poolse](#) aggiorna il cluster dopo aver impostato [LoginNodes/Poolse/Counts](#) su 0. Quindi, esegui il comando seguente dal nodo principale del cluster.

```
sudo /opt/parallelcluster/scripts/directory_service/  
update_directory_service_password.sh
```

Per un altro esempio, vedi anche [Integrazione di Active Directory](#).

Accedere a un cluster integrato con un dominio AD

Se è stata abilitata la funzionalità di integrazione del dominio Active Directory (AD), l'autenticazione tramite password viene abilitata sul nodo principale del cluster. La home directory di un utente AD viene creata al primo accesso utente al nodo principale o alla prima volta che `sudo-user` passa all'utente AD sul nodo principale.

L'autenticazione tramite password non è abilitata per i nodi di calcolo del cluster. Gli utenti AD devono accedere ai nodi di calcolo con chiavi SSH.

Per impostazione predefinita, le chiavi SSH vengono configurate nella `/${HOME}/.ssh` directory utente AD al primo accesso SSH al nodo principale. Questo comportamento può essere disabilitato impostando la proprietà [DirectoryService/GenerateSshKeysForUsers](#) boolean su `false` nella configurazione del cluster. Per impostazione predefinita, [DirectoryService/GenerateSshKeysForUsers](#) è impostato su `true`.

Se un' AWS ParallelCluster applicazione richiede SSH senza password tra i nodi del cluster, assicurati che le chiavi SSH siano impostate correttamente nella home directory dell'utente.

AWS Managed Microsoft AD le password scadono dopo 42 giorni. Per ulteriori informazioni, consulta [Gestire le politiche relative alle password AWS Managed Microsoft AD](#) nella Guida all'Directory Service amministrazione. Se la password scade, deve essere reimpostata per ripristinare l'accesso al cluster. Per ulteriori informazioni, consulta [Come reimpostare una password utente e le password scadute](#).

Note

Se la funzionalità di integrazione AD non funziona come previsto, i log SSSD possono fornire informazioni diagnostiche utili per la risoluzione del problema. Questi log si trovano nella `/var/log/sss` directory dei nodi del cluster. Per impostazione predefinita, vengono archiviati anche nel gruppo di CloudWatch log Amazon di un cluster.

Per ulteriori informazioni, consulta [Risoluzione dei problemi di integrazione multiutente con Active Directory](#).

Esecuzione di processi MPI

Come suggerito in SchedMD, avviate i job MPI utilizzando come metodo di avvio MPI. Slurm Per ulteriori informazioni, consultate la documentazione ufficiale o la [Slurmdocumentazione](#) ufficiale della libreria MPI in uso.

Ad esempio, nella [documentazione ufficiale di IntelMPI](#), si apprende che quando si esegue un job StarCCM, è necessario impostarlo Slurm come orchestratore di processi esportando la variabile di ambiente. `I_MPI_HYDRA_BOOTSTRAP=slurm`

Note

Problema noto

Nel caso in cui l'applicazione MPI si basi su SSH come meccanismo per generare job MPI, è possibile incorrere in un [bug noto](#) in Slurm che causa la risoluzione errata del nome utente della directory a «nessuno».

Configura l'applicazione per utilizzarla Slurm come metodo di avvio MPI o consulta la sezione Risoluzione dei problemi per ulteriori dettagli e possibili soluzioni alternative. [Problemi noti relativi alla risoluzione dei nomi utente](#)

Esempio di configurazioni di cluster AWS Managed Microsoft AD LDAP (S)

AWS ParallelCluster supporta l'accesso di più utenti mediante l'integrazione con un protocollo LDAP (Lightweight Directory Access Protocol) o LDAP AWS Directory Service over (LDAPS). TLS/SSL

Gli esempi seguenti mostrano come creare configurazioni di cluster da integrare con un AWS Managed Microsoft AD protocollo LDAP (S).

AWS Managed Microsoft AD su LDAPS con verifica del certificato

È possibile utilizzare questo esempio per integrare il cluster con un AWS Managed Microsoft AD protocollo LDAPS, con verifica del certificato.

Definizioni specifiche per una configurazione AWS Managed Microsoft AD tramite LDAPS con certificati:

- [DirectoryService/LdapTlsReqCert](#) deve essere impostato su `hard` (impostazione predefinita) per LDAPS con verifica del certificato.
- [DirectoryService/LdapTlsCaCert](#) deve specificare il percorso del certificato di autorità (CA).

Il certificato CA è un pacchetto di certificati che contiene i certificati dell'intera catena CA che ha emesso certificati per i controller di dominio AD.

Il certificato e i certificati CA devono essere installati nei nodi del cluster.

- I nomi host dei controller devono essere specificati per [DirectoryService/DomainAddr](#), non per gli indirizzi IP.
- [DirectoryService](#)/la [DomainReadOnlyUser](#) sintassi deve essere la seguente:

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Esempio di file di configurazione del cluster per l'utilizzo di AD su LDAPS:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
  CustomActions:
    OnNodeConfigured:
      Script: s3://&example-s3-bucket;/scripts/pcluster-dub-msad-ldaps.post.sh
  Scheduling:
```

```

Scheduler: slurm
SlurmQueues:
  - Name: queue1
    ComputeResources:
      - Name: t2micro
        InstanceType: t2.micro
        MinCount: 1
        MaxCount: 10
    Networking:
      SubnetIds:
        - subnet-abcdef01234567890
    Iam:
      AdditionalIamPolicies:
        - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
    CustomActions:
      OnNodeConfigured:
        Script: s3://&example-s3-bucket;/scripts/pcluster-dub-msad-ldaps.post.sh
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://win-abcdef01234567890.corp.example.com,ldaps://win-
abcdef01234567890.corp.example.com
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsCaCert: /etc/ldap/cacerts/corp.example.com.bundleca.cer
  LdapTlsReqCert: hard

```

Aggiungi certificati e configura i controller di dominio nello script post-installazione:

```

#!/bin/bash
set -e

AD_CERTIFICATE_S3_URI="s3://amzn-s3-demo-bucket/bundle/corp.example.com.bundleca.cer"
AD_CERTIFICATE_LOCAL="/etc/ldap/cacerts/corp.example.com.bundleca.cer"

AD_HOSTNAME_1="win-abcdef01234567890.corp.example.com"
AD_IP_1="192.0.2.254"

AD_HOSTNAME_2="win-abcdef01234567890.corp.example.com"
AD_IP_2="203.0.113.225"

# Download CA certificate
mkdir -p $(dirname "${AD_CERTIFICATE_LOCAL}")

```

```
aws s3 cp "${AD_CERTIFICATE_S3_URI}" "${AD_CERTIFICATE_LOCAL}"
chmod 644 "${AD_CERTIFICATE_LOCAL}"

# Configure domain controllers reachability
echo "${AD_IP_1} ${AD_HOSTNAME_1}" >> /etc/hosts
echo "${AD_IP_2} ${AD_HOSTNAME_2}" >> /etc/hosts
```

È possibile recuperare i nomi host dei controller di dominio dalle istanze aggiunte al dominio, come illustrato negli esempi seguenti.

Dall'istanza di Windows

```
$ nslookup 192.0.2.254
```

```
Server: corp.example.com
Address: 192.0.2.254

Name: win-abcdef01234567890.corp.example.com
Address: 192.0.2.254
```

Dall'istanza Linux

```
$ nslookup 192.0.2.254
```

```
192.0.2.254.in-addr.arpa name = corp.example.com
192.0.2.254.in-addr.arpa name = win-abcdef01234567890.corp.example.com
```

AWS Managed Microsoft AD su LDAPS senza verifica del certificato

È possibile utilizzare questo esempio per integrare il cluster con un protocollo LDAPS AWS Managed Microsoft AD tramite LDAPS, senza verifica del certificato.

Definizioni specifiche per una configurazione AWS Managed Microsoft AD tramite LDAPS senza verifica del certificato:

- [DirectoryService/LdapTlsReqCert](#) deve essere impostato su `never`
- È possibile specificare i nomi host o gli indirizzi IP dei controller per [DirectoryService/DomainAddr](#).
- [DirectoryService](#)/la [DomainReadOnlyUser](#) sintassi deve essere la seguente:


```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Esempio di file di configurazione del cluster da utilizzare AWS Managed Microsoft AD su LDAPS senza verifica del certificato:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://203.0.113.225,ldaps://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsReqCert: never
```

Best practice

Le seguenti sezioni forniscono le migliori pratiche per l'utilizzo AWS ParallelCluster, tra cui avvisi sulle prestazioni di rete e sul budget. Se riscontri problemi anche se segui queste procedure consigliate, consulta [AWS ParallelCluster risoluzione dei problemi](#) le possibili soluzioni.

Procedure consigliate: selezione del tipo di istanza del nodo principale

Anche se il nodo principale non esegue un processo, le sue funzioni e il suo dimensionamento sono fondamentali per le prestazioni complessive del cluster. Quando scegli il tipo di istanza da utilizzare per il tuo nodo principale, considera le seguenti caratteristiche:

Dimensioni del cluster: il nodo principale orchestra la logica di scalabilità del cluster ed è responsabile del collegamento di nuovi nodi allo scheduler. Per scalare verso l'alto e verso il basso un cluster con un numero elevato di nodi, fornisci al nodo principale una capacità di elaborazione aggiuntiva.

File system condivisi: quando utilizzi file system condivisi, scegli un tipo di istanza con una larghezza di banda di rete sufficiente e una larghezza di banda Amazon EBS sufficiente per gestire i flussi di lavoro. Assicurati che il nodo principale sia in grado di esporre directory di server NFS sufficienti per il cluster e gestire gli artefatti che devono essere condivisi tra i nodi di calcolo e il nodo principale.

Migliori pratiche: prestazioni di rete

Le prestazioni di rete sono fondamentali per le applicazioni HPC (High Performance Computing). Senza prestazioni di rete affidabili, queste applicazioni non possono funzionare come previsto. Per ottimizzare le prestazioni della rete, prendete in considerazione le seguenti best practice.

- **Gruppo di posizionamento:** se utilizzi Slurm, valuta la possibilità di configurare ogni Slurm coda per utilizzare un gruppo di posizionamento del cluster. Il gruppo di posizionamento di un cluster è un raggruppamento logico di istanze all'interno di una singola zona di disponibilità. Per ulteriori informazioni, consulta i [gruppi di collocamento](#) nella Amazon EC2 User Guide. Puoi specificare a [PlacementGroup](#) nella [Networking](#) sezione della coda, ogni risorsa di calcolo viene assegnata al gruppo di posizionamento della coda. Quando si specifica a [PlacementGroup](#) nella [Networking](#) sezione della risorsa di calcolo, quella risorsa di calcolo specifica viene assegnata a quel gruppo di posizionamento. La specifica del gruppo di posizionamento delle risorse di calcolo ha la precedenza sulla specifica della coda per la risorsa di calcolo. Per ulteriori informazioni, vedere [SlurmQueues//e/Networking/PlacementGroup](#). [SlurmQueuesComputeResourcesNetworkingPlacementGroup](#)

```
Networking:
  PlacementGroup:
    Enabled: true
    Id: your-placement-group-name
```

In alternativa, AWS ParallelCluster crea un gruppo di collocamento per te.

```
Networking:
  PlacementGroup:
    Enabled: true
```

A partire dalla AWS ParallelCluster versione 3.3.0, la creazione e la gestione dei gruppi di collocamento sono state modificate. Quando specificate il gruppo di posizionamento da abilitare, senza un `name oId`, nella coda, a ciascuna risorsa di calcolo viene assegnato un proprio gruppo di posizionamento gestito, anziché un gruppo gestito per l'intera coda. Questo aiuta a ridurre gli errori di capacità insufficiente. Se è necessario disporre di un gruppo di posizionamento per l'intera coda, è possibile utilizzare un gruppo di posizionamento denominato.

[SlurmQueues/Networking/PlacementGroup/Name](#) è stato aggiunto come alternativa preferita a [SlurmQueues//NetworkingPlacementGroup/Id](#).

Per ulteriori informazioni, consulta [Networking](#).

- Rete avanzata: prendete in considerazione la scelta di un tipo di istanza che supporti la rete avanzata. Questa raccomandazione si applica a tutte le [istanze di generazione attuale](#). Per ulteriori informazioni, consulta la sezione [Enhanced Networking on Linux](#) nella Amazon EC2 User Guide.
- Elastic Fabric Adapter: per supportare livelli elevati di comunicazione scalabile da istanza a istanza, prendi in considerazione la possibilità di scegliere interfacce di rete EFA per la tua rete. L'hardware di bypass del sistema operativo (OS) personalizzato dell'EFA migliora le comunicazioni da istanza a istanza con l'elasticità e la flessibilità on-demand di Cloud AWS. È possibile configurare ogni coda da utilizzare. Slurm [ComputeResourceEfa](#) Per ulteriori informazioni sull'utilizzo di EFA con AWS ParallelCluster, vedere [Elastic Fabric Adapter](#)

```
ComputeResources:
  - Name: your-compute-resource-name
    Efa:
      Enabled: true
```

Per ulteriori informazioni su EFA, consulta [Elastic Fabric Adapter](#) nella Amazon EC2 User Guide for Linux Instances.

- Larghezza di banda dell'istanza: la larghezza di banda varia in base alla dimensione dell'istanza. Per informazioni sui diversi tipi di istanze, consulta le [istanze ottimizzate per Amazon EBS](#) e i tipi di [volume Amazon EBS nella](#) Amazon User Guide. EC2

Migliori pratiche: avvisi sul budget

Per gestire i costi delle risorse in AWS ParallelCluster, ti consigliamo di utilizzare Budget AWS le azioni per creare un budget. È inoltre possibile creare avvisi relativi alle soglie di budget definite per AWS risorse selezionate. Per ulteriori informazioni, consulta [Configurazione di un'azione di budget nella Guida](#) per l'utente Budget AWS. Allo stesso modo, puoi anche utilizzare Amazon CloudWatch per creare un allarme di fatturazione. Per ulteriori informazioni, consulta [Creazione di un allarme di fatturazione per il monitoraggio dei costi di AWS stimati](#).

Migliori pratiche: spostare un cluster in una nuova versione AWS ParallelCluster secondaria o patch

Attualmente ogni versione AWS ParallelCluster secondaria è autonoma insieme alla relativa `pccluster` CLI. Per spostare un cluster in una nuova versione secondaria o patch, è necessario ricreare il cluster utilizzando la CLI della nuova versione.

Per ottimizzare il processo di spostamento di un cluster in una nuova versione secondaria o patch, si consiglia di effettuare le seguenti operazioni:

- Salva i dati personali in volumi esterni creati all'esterno del cluster, come Amazon EFS e FSx for Lustre. In questo modo, è possibile spostare facilmente i dati da un cluster all'altro in futuro.
- Crea sistemi di storage condivisi utilizzando i seguenti tipi. È possibile creare questi sistemi utilizzando AWS CLI o Console di gestione AWS.
 - [SharedStorage](#) / [EbsSettings](#) / [VolumeId](#)
 - [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)
 - [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)

Definire un file system o un volume in una configurazione del cluster come file system o volume esistente. In questo modo, vengono conservati quando si elimina il cluster e possono essere collegati a un nuovo cluster.

Ti consigliamo di utilizzare i file system Amazon EFS o FSx for Lustre. Entrambi questi sistemi possono essere collegati a più cluster contemporaneamente. Inoltre, è possibile collegare uno di questi sistemi a un nuovo cluster prima di eliminare il cluster esistente.

- Utilizza [azioni bootstrap personalizzate](#) per personalizzare le istanze anziché utilizzare un'AMI personalizzata. Se invece utilizzi un'AMI personalizzata, devi eliminare e ricreare quell'AMI per ogni nuova versione rilasciata.

- Ti consigliamo di applicare i consigli precedenti nella seguente sequenza:
 1. Aggiorna la configurazione del cluster esistente per utilizzare le definizioni del file system esistenti.
 2. Verifica la `pcluster` versione e aggiornala se necessario.
 3. Crea e testa il nuovo cluster. Quando testate il nuovo cluster, controllate quanto segue:
 - Assicuratevi che i tuoi dati siano disponibili nel nuovo cluster.
 - Assicuratevi che l'applicazione funzioni nel nuovo cluster.
 4. Dopo che il nuovo cluster è stato completamente testato e reso operativo e non è più necessario il cluster esistente, eliminatelo.

Passaggio da AWS ParallelCluster 2.x a 3.x

Le sezioni seguenti descrivono cosa succede quando si passa dalla AWS ParallelCluster 2.x alla 3.x, incluse le modifiche da una versione all'altra.

Azioni Bootstrap personalizzate

Con AWS ParallelCluster 3, è possibile specificare diversi script di azioni bootstrap personalizzati per il nodo principale e i nodi di calcolo utilizzando `OnNodeStart` (`pre_install` nella AWS ParallelCluster versione 2) e `OnNodeConfigured` (`post_install` nella AWS ParallelCluster versione 2) i parametri nelle sezioni e/. [HeadNodeSchedulingSlurmQueues](#) Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

Gli script di azioni bootstrap personalizzati sviluppati per AWS ParallelCluster 2 devono essere adattati per essere utilizzati in 3: AWS ParallelCluster

- Non è consigliabile utilizzare `/etc/parallelcluster/cfnconfig` e `cfn_node_type` distinguere tra nodi principali e nodi di calcolo. Ti consigliamo invece di specificare due script diversi in e/. [HeadNodeSchedulingSlurmQueues](#)
- Se preferite continuare a caricarlo `/etc/parallelcluster/cfnconfig` per utilizzarlo nello script di azioni bootstrap, tenete presente che il valore di `cfn_node_type` è cambiato da "MasterServer" a "HeadNode" (vedi: [Lingua inclusiva](#)).
- Su AWS ParallelCluster 2, il primo argomento di input per gli script di azione bootstrap era l'URL S3 dello script ed era riservato. In AWS ParallelCluster 3, solo gli argomenti configurati nella configurazione vengono passati agli script.

 Warning

L'utilizzo di variabili interne fornite tramite il `/etc/parallelcluster/cfnconfig` file non è ufficialmente supportato. Questo file potrebbe essere rimosso come parte di future release.

AWS ParallelCluster 2.x e 3.x utilizzano una sintassi diversa del file di configurazione

AWS ParallelCluster La configurazione 3.x utilizza la sintassi YAML. Il riferimento completo può essere trovato all'indirizzo. [File di configurazione](#)

Oltre a richiedere un formato di file YAML, nella versione 3.x sono state aggiornate diverse sezioni di configurazione, impostazioni e valori dei parametri. AWS ParallelCluster In questa sezione, annotiamo le principali modifiche alla AWS ParallelCluster configurazione insieme a side-by-side esempi che illustrano queste differenze tra ciascuna versione di AWS ParallelCluster

Esempio di configurazione di code di pianificazione multiple con hyperthreading abilitato e disabilitato

AWS ParallelCluster 2:

```
[cluster default]
queue_settings = ht-enabled, ht-disabled
...

[queue ht-enabled]
compute_resource_settings = ht-enabled-i1
disable_hyperthreading = false

[queue ht-disabled]
compute_resource_settings = ht-disabled-i1
disable_hyperthreading = true

[compute_resource ht-enabled-i1]
instance_type = c5n.18xlarge
[compute_resource ht-disabled-i1]
instance_type = c5.xlarge
```

AWS ParallelCluster 3:

```

...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: ht-enabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-enabled-i1
          DisableSimultaneousMultithreading: true
          InstanceType: c5n.18xlarge
    - Name: ht-disabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-disabled-i1
          DisableSimultaneousMultithreading: false
          InstanceType: c5.xlarge

```

Esempio di novità FSx per la configurazione del file system Lustre

AWS ParallelCluster 2:

```

[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
storage_capacity = 1200
imported_file_chunk_size = 1024
import_path = s3://amzn-s3-demo-bucket
export_path = s3://amzn-s3-demo-bucket/export_dir
weekly_maintenance_start_time = 3:02:30
deployment_type = PERSISTENT_1
data_compression_type = LZ4

```

AWS ParallelCluster 3:

```

...

```

```
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      ImportedFileChunkSize: 1024
      ImportPath: s3://amzn-s3-demo-bucket
      ExportPath: s3://amzn-s3-demo-bucket/export_dir
      WeeklyMaintenanceStartTime: "3:02:30"
      DeploymentType: PERSISTENT_1
      DataCompressionType: LZ4
```

Esempio di configurazione di cluster che monta un file system esistente FSx per Lustre

AWS ParallelCluster 2:

```
[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
fsx_fs_id = fsx_fs_id
```

AWS ParallelCluster 3:

```
...
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      FileSystemId: fsx_fs_id
```

Esempio di cluster con lo stack software Intel HPC Platform Specification

AWS ParallelCluster 2:

```
[cluster default]
enable_intel_hpc_platform = true
...
```


AWS ParallelCluster 3:

```
...
AdditionalPackages:
  IntelSoftware:
    IntelHpcPlatform: true
```

Note:

- L'installazione del software Intel HPC Platform Specification è soggetta ai termini e alle condizioni del [Contratto di licenza per l'utente finale Intel](#) applicabile.

Esempi di configurazioni IAM personalizzate, tra cui: profilo dell'istanza, ruolo dell'istanza, politiche aggiuntive per le istanze e il ruolo per le funzioni lambda associate al cluster

AWS ParallelCluster 2:

```
[cluster default]
additional_iam_policies = arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess,arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
ec2_iam_role = ec2_iam_role
iam_lambda_role = lambda_iam_role
...
```

AWS ParallelCluster 3:

```
...
Iam:
  Roles:
    CustomLambdaResources: lambda_iam_role
HeadNode:
  ...
  Iam:
    InstanceRole: ec2_iam_role
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
    Iam:
      InstanceProfile: iam_instance_profile
```

```
- Name: queue2
...
Iam:
  AdditionalIamPolicies:
    - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
    - Policy: arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
```

Note:

- Per AWS ParallelCluster 2, le impostazioni IAM vengono applicate a tutte le istanze di un cluster e non `additional_iam_policies` possono essere utilizzate insieme a `ec2_iam_role`
- Per AWS ParallelCluster 3, puoi avere impostazioni IAM diverse per i nodi principali e di calcolo e persino specificare impostazioni IAM diverse per ogni coda di calcolo.
- Per AWS ParallelCluster 3, puoi utilizzare un profilo di istanza IAM come alternativa a un ruolo IAM. `InstanceProfileInstanceRole` e non `AdditionalIamPolicies` possono essere configurati insieme.

Esempio di azioni bootstrap personalizzate

AWS ParallelCluster 2:

```
[cluster default]
s3_read_resource = arn:aws:s3:::amzn-s3-demo-bucket/*
pre_install = s3://amzn-s3-demo-bucket/scripts/pre_install.sh
pre_install_args = 'R curl wget'
post_install = s3://amzn-s3-demo-bucket/scripts/post_install.sh
post_install_args = "R curl wget"
...
```

AWS ParallelCluster 3:

```
...
HeadNode:
  ...
  CustomActions:
    OnNodeStart:
      Script: s3://amzn-s3-demo-bucket/scripts/pre_install.sh
      Args:
        - R
        - curl
        - wget
```

```

OnNodeConfigured:
  Script: s3://amzn-s3-demo-bucket/scripts/post_install.sh
  Args: ['R', 'curl', 'wget']
Iam:
  S3Access:
    - BucketName: amzn-s3-demo-bucket
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
  CustomActions:
    OnNodeStart:
      Script: s3://amzn-s3-demo-bucket/scripts/pre_install.sh
      Args: ['R', 'curl', 'wget']
    OnNodeConfigured:
      Script: s3://amzn-s3-demo-bucket/scripts/post_install.sh
      Args: ['R', 'curl', 'wget']
Iam:
  S3Access:
    - BucketName: amzn-s3-demo-bucket

```

Esempio di cluster con accesso in lettura e scrittura alle risorse del bucket S3

AWS ParallelCluster 2:

```

[cluster default]
s3_read_resource = arn:aws:s3:::amzn-s3-demo-bucket/read_only/*
s3_read_write_resource = arn:aws:s3:::amzn-s3-demo-bucket/read_and_write/*
...

```

AWS ParallelCluster 3:

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: amzn-s3-demo-bucket
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: amzn-s3-demo-bucket
        KeyName: read_and_write/

```

```

    EnableWriteAccess: True
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
  Iam:
    S3Access:
      - BucketName: amzn-s3-demo-bucket
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: amzn-s3-demo-bucket
        KeyName: read_and_write/
        EnableWriteAccess: True

```

Lingua inclusiva

AWS ParallelCluster 3 usa le parole «head node» nei punti in cui «master» è stato usato in AWS ParallelCluster 2. Questo include gli output seguenti:

- Variabile esportata nell'ambiente di AWS Batch lavoro modificata: da MASTER_IP a PCLUSTER_HEAD_NODE_IP.
- Tutti gli CloudFormation output sono cambiati da aMaster*. HeadNode*
- Tutti NodeType i tag sono cambiati da Master a HeadNode

Supporto Scheduler

AWS ParallelCluster 3.x non supporta gli scheduler Son of Grid Engine (SGE) e Torque.

I AWS Batch comandi `awsbhosts`, `awsbkill`, `awsbout`, `awsbqueues`, `awsbstat`, e `awsbsub` sono distribuiti come pacchetto `aws-parallelcluster-awsbatch-cli` PyPI separato. Questo pacchetto viene installato dal AWS ParallelCluster nodo principale. È ancora possibile utilizzare questi AWS Batch comandi dal nodo principale del cluster. Tuttavia, se desideri utilizzare AWS Batch i comandi da una posizione diversa dal nodo principale, devi prima installare il pacchetto `aws-parallelcluster-awsbatch-cli` PyPI.

AWS ParallelCluster CLI

L'interfaccia AWS ParallelCluster a riga di comando (CLI) è stata modificata. La nuova sintassi è descritta in [AWS ParallelCluster Comandi CLI](#). Il formato di output per la CLI è una stringa [JSON](#).

Configurazione di un nuovo cluster

Il `pcluster configure` comando include parametri diversi in AWS ParallelCluster 3 rispetto a AWS ParallelCluster 2. Per ulteriori informazioni, consulta [pcluster configure](#).

Si noti inoltre che la sintassi del file di configurazione è cambiata rispetto a AWS ParallelCluster 2. Per un riferimento completo delle impostazioni di configurazione del cluster, vedere [File di configurazione del cluster](#).

Creazione di un nuovo cluster

AWS ParallelCluster Il `pcluster create` comando di 2 è stato sostituito dal [pcluster create-cluster](#) comando.

Nota che il comportamento predefinito in AWS ParallelCluster 2.x, senza l'-nw opzione, è quello di attendere gli eventi di creazione del cluster, mentre il comando AWS ParallelCluster 3.x ritorna immediatamente. L'avanzamento della creazione del cluster può essere monitorato utilizzando [pcluster describe-cluster](#)

Un file di configurazione a AWS ParallelCluster 3 dimensioni contiene una singola definizione di cluster, quindi il `-t` parametro non è più necessario.

Di seguito è riportato un esempio di file di configurazione.

```
# AWS ParallelCluster v2
$ pcluster create \
  -r REGION \
  -c V2_CONFIG_FILE \
  -nw \
  -t CLUSTER_TEMPLATE \
  CLUSTER_NAME

# AWS ParallelCluster v3
$ pcluster create-cluster \
  --region REGION \
  --cluster-configuration V3_CONFIG_FILE \
  --cluster-name CLUSTER_NAME
```

Elenco dei cluster

Il comando `pcluster list` AWS ParallelCluster 2.x deve essere sostituito con [pcluster list-clusters](#) command.

Nota: è necessaria la CLI AWS ParallelCluster v2 per elencare i cluster creati con le versioni 2.x di AWS ParallelCluster. Scopri [Installa AWS ParallelCluster in un ambiente virtuale \(consigliato\)](#) come installare più versioni di Using Virtual Environments. AWS ParallelCluster

```
# AWS ParallelCluster v2
$ pcluster list -r REGION

# AWS ParallelCluster v3
$ pcluster list-clusters --region REGION
```

Avvio e arresto di un cluster

I comandi `pcluster start` e `pcluster stop` AWS ParallelCluster 2.x devono essere sostituiti con [pcluster update-compute-fleet](#) comandi.

Avvio di una flotta di elaborazione:

```
# AWS ParallelCluster v2
$ pcluster start \
  -r REGION \
  CLUSTER_NAME

# AWS ParallelCluster v3 - Slurm fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status START_REQUESTED

# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status ENABLED
```

Arresto di una flotta di elaborazione:

```
# AWS ParallelCluster v2
$ pcluster stop \
  -r REGION \
  CLUSTER_NAME

# AWS ParallelCluster v3 - Slurm fleets
```

```
$ pcluster update-compute-fleet \  
  --region REGION \  
  --cluster-name CLUSTER_NAME \  
  --status STOP_REQUESTED  
  
# AWS ParallelCluster v3 - AWS Batch fleets  
$ pcluster update-compute-fleet \  
  --region REGION \  
  --cluster-name CLUSTER_NAME \  
  --status DISABLED
```

Connessione a un cluster

Il comando `pcluster ssh` AWS ParallelCluster 2.x ha nomi di parametri diversi in AWS ParallelCluster 3.x. Consultare [pcluster ssh](#).

Connessione a un cluster:

```
# AWS ParallelCluster v2  
$ pcluster ssh \  
  -r REGION \  
  CLUSTER_NAME \  
  -i ~/.ssh/id_rsa  
  
# AWS ParallelCluster v3  
$ pcluster ssh \  
  --region REGION \  
  --cluster-name CLUSTER_NAME \  
  -i ~/.ssh/id_rsa
```

Aggiornamento della configurazione IMDS

A partire dalla versione 3.0.0, AWS ParallelCluster ha introdotto il supporto per limitare l'accesso all'IMDS del nodo principale (e alle credenziali del profilo di istanza) a un sottoinsieme di superutenti, per impostazione predefinita. Per ulteriori informazioni, consulta [Proprietà Imds](#).

Usando AWS ParallelCluster

Argomenti

- [AWS ParallelCluster UI](#)
- [AWS Lambda Configurazione VPC in AWS ParallelCluster](#)
- [AWS Identity and Access Management autorizzazioni in AWS ParallelCluster](#)
- [Configurazioni di rete](#)
- [Nodi di accesso forniti da AWS ParallelCluster](#)
- [Azioni bootstrap personalizzate](#)
- [Lavorare con Amazon S3](#)
- [Utilizzo di Istanze spot](#)
- [Scheduler supportati da AWS ParallelCluster](#)
- [Archiviazione condivisa](#)
- [AWS ParallelCluster risorse e etichettatura](#)
- [Monitoraggio AWS ParallelCluster e registri](#)
- [AWS CloudFormation risorsa personalizzata](#)
- [Elastic Fabric Adapter](#)
- [Abilitare Intel MPI](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster per Terraform](#)
- [Connect ai nodi principali e di accesso tramite Amazon DCV](#)
- [Uso di pcluster update-cluster](#)
- [AWS ParallelCluster Personalizzazione AMI](#)
- [Avvio di istanze con On-Demand Capacity Reservations \(ODCR\)](#)
- [Avvia istanze con Capacity Blocks \(CB\)](#)
- [Patch AMI e sostituzione delle EC2 istanze Amazon](#)
- [Sistemi operativi](#)

AWS ParallelCluster UI

L'AWS ParallelCluster interfaccia utente (PCUI) è un'interfaccia utente basata sul Web che funge da dashboard per la creazione, il monitoraggio e la gestione dei cluster. Puoi installare e accedere al PCUI nel tuo Account AWS. Il PCUI viene aggiunto con la AWS ParallelCluster versione 3.5.0.

Per installare il PCUI e iniziare, consulta e. [Installazione del PCUI](#) [Configurazione e creazione di un cluster con PCUI](#)

The screenshot displays the AWS ParallelCluster user interface. The top navigation bar shows the AWS ParallelCluster logo, the user email 'user@domain.com', and the region 'eu-west-1'. The left sidebar contains links for 'Clusters', 'Images', and 'Users', with 'Clusters' being the active section. The main content area is titled 'Clusters (2)' and includes buttons for 'Shell', 'DCV', 'Stop fleet', 'Actions', and 'Create cluster'. Below this is a search bar and a table listing the clusters. The table has columns for Name, Status, and Version. Two clusters are listed: 'hpc-cluster-1' with status 'CREATE COMPLETE' and version '3.5.0', and 'hpc-cluster-2' with status 'DELETE IN PROGRESS' and version '3.5.0'. Below the table, the details for 'Cluster: hpc-cluster-1' are shown, including tabs for 'Details', 'Instances', 'Storage', 'Job scheduling', and 'Stack events'. The 'Details' tab is active, showing properties such as Cluster configuration, SSH command, EC2 Instance Connect, Cluster status (CREATE COMPLETE), Compute fleet status (RUNNING), Version (3.5.0), Region (eu-west-1), Created time (March 10, 2023 at 09:39 (UTC+1:00)), and Latest update time (March 10, 2023 at 09:39 (UTC+1:00)).

Name	Status	Version
hpc-cluster-1	CREATE COMPLETE	3.5.0
hpc-cluster-2	DELETE IN PROGRESS	3.5.0

Cluster: hpc-cluster-1		
Details	Instances	Storage
Properties		
Cluster configuration VIEW	Cluster status CREATE COMPLETE	Region eu-west-1
SSH command Info ssh ec2-user@54.78.245.22	Compute fleet status RUNNING	Created time March 10, 2023 at 09:39 (UTC+1:00)
EC2 Instance Connect Info mssh -r eu-west-1 ec2-user@i-0b14dc1a2f5dc048e	Version 3.5.0	Latest update time March 10, 2023 at 09:39 (UTC+1:00)

Il PCUI supporta le seguenti funzionalità:

- Visualizza quanto segue:
 - L'elenco dei cluster che hai creato Account AWS con AWS ParallelCluster.
 - Lo stato e i dettagli disponibili per i cluster elencati.

- CloudFormation raggruppa gli eventi e AWS ParallelCluster i log che puoi utilizzare per il monitoraggio.
- Lo stato dei job in esecuzione sui cluster.
- L'elenco di immagini personalizzate che puoi utilizzare per creare cluster.
- L'elenco delle immagini ufficiali utilizzate dall'interfaccia utente per creare cluster.
- L'elenco degli utenti che hanno accesso al PCUI. È possibile aggiungere e rimuovere utenti.
- Fornisce step-by-step indicazioni per creare e modificare (aggiornare) un cluster e selezionare le funzionalità del cluster supportate da aggiungere, modificare o rimuovere. I campi di input inaccessibili non possono essere modificati per la configurazione del cluster in corso di modifica. È possibile eseguire una convalida a secco della configurazione del cluster prima della distribuzione del cluster.
- Presenta collegamenti diretti alla shell per accedere al nodo principale nella vista Clusters. Scegli Aggiungi sessione SSM durante la step-by-step guida per aggiungere l'accesso diretto alla shell e la policy SSM Managed Instance Core sul nodo principale.

Quando utilizzi il PCUI per creare e gestire i cluster, considera quanto segue:

- È possibile creare e modificare cluster o creare immagini solo con la stessa AWS ParallelCluster versione utilizzata per creare il PCUI. I cluster o le immagini delle versioni precedenti possono essere visualizzati solo. Se gestisci più versioni di cluster e immagini, ti consigliamo di creare un'istanza PCUI per supportare ciascuna versione.
- Il PCUI è progettato per rispecchiare la funzionalità `pcluster CLI`. Ci sono alcune differenze. Se ti allinei alla step-by-step guida, stai utilizzando tutte le funzionalità supportate. Prima della distribuzione, è possibile modificare manualmente la configurazione del cluster o dell'immagine. In tal caso, ti consigliamo di convalidare la configurazione scegliendo Dry run per verificare che le modifiche siano completamente supportate.

Note

PCUI non supporta. AWS Batch

AWS Lambda Configurazione VPC in AWS ParallelCluster

AWS ParallelCluster utilizza AWS Lambda per eseguire operazioni durante il ciclo di vita del cluster. Una [AWS Lambda funzione viene sempre eseguita in un VPC](#) di proprietà del servizio Lambda. Questa funzione Lambda può anche essere connessa a sottoreti private in un cloud privato virtuale (VPC) per accedere a risorse private.

Note

Le funzioni Lambda non possono connettersi direttamente a un VPC con la tenancy dell'istanza dedicata. Per connetterti alle risorse in un VPC dedicato, trasferisci il VPC dedicato a un secondo VPC con una tenancy predefinita in grado di connettersi a un VPC dedicato.

Per ulteriori informazioni, consulta [Dedicated Instances](#) nella Amazon EC2 User Guide for Linux Instances e [Come posso connettere una funzione Lambda a un VPC dedicato?](#) dal Knowledge Center.AWS

Le funzioni Lambda create da AWS ParallelCluster possono essere collegate a un VPC privato. Queste funzioni Lambda devono accedere. Servizi AWS Puoi fornire l'accesso tramite Internet o endpoint VPC utilizzando i seguenti metodi.

- Accesso a Internet

Per accedere a Internet e Servizi AWS, una funzione Lambda richiede la traduzione degli indirizzi di rete (NAT). Indirizza il traffico in uscita dalla sottorete privata a un [gateway NAT in](#) una sottorete pubblica.

- Endpoint VPC

Diversi AWS servizi offrono [endpoint VPC](#). Puoi utilizzare gli endpoint VPC per connetterti da Servizi AWS un VPC che non dispone di accesso a Internet. [Per visualizzare l'elenco degli endpoint AWS ParallelCluster VPC, consulta Networking.](#)

 Note

Ogni combinazione di sottoreti e gruppi di sicurezza deve consentire l'accesso all' Servizi AWS utilizzo di uno di questi metodi. Le sottoreti e i gruppi di sicurezza devono trovarsi nello stesso VPC.

Per ulteriori informazioni, consulta gli [endpoint VPC](#) nella Amazon Virtual Private Cloud User Guide e l'[accesso a Internet e ai servizi per le funzioni connesse a VPC](#) nella Developer Guide.AWS Lambda

Per configurare l'uso delle funzioni Lambda e VPCs, vedi [DeploymentSettings/LambdaFunctionsVpcConfig](#) per i cluster o [DeploymentSettings/LambdaFunctionsVpcConfig](#) per le immagini.

AWS Identity and Access Management autorizzazioni in AWS ParallelCluster

AWS ParallelCluster utilizza le autorizzazioni IAM per controllare l'accesso alle risorse durante la creazione e la gestione dei cluster.

Per creare e gestire i cluster in un AWS account, sono AWS ParallelCluster necessarie autorizzazioni a due livelli:

- Autorizzazioni richieste dall'`pcluster`utente per richiamare i comandi `pcluster` CLI per la creazione e la gestione dei cluster.
- Autorizzazioni richieste dalle risorse del cluster per eseguire le azioni del cluster.

AWS ParallelCluster utilizza un [profilo e un ruolo dell'istanza Amazon EC2 per fornire le autorizzazioni](#) per le risorse del cluster. Per gestire le autorizzazioni delle risorse del cluster, sono necessarie AWS ParallelCluster anche le autorizzazioni per le risorse IAM. Per ulteriori informazioni, consulta [AWS ParallelCluster esempi di politiche utente per la gestione delle risorse IAM](#).

`pcluster`gli utenti richiedono le autorizzazioni IAM per utilizzare la `pcluster` CLI per creare e gestire un cluster e le relative risorse. Queste autorizzazioni sono incluse nelle policy IAM che possono essere aggiunte a un utente o a un ruolo. Per ulteriori informazioni sui ruoli IAM, consulta [Creazione di un ruolo utente](#) nella Guida per l'AWS Identity and Access Management utente.

È possibile utilizzare anche [AWS ParallelCluster parametri di configurazione per gestire le autorizzazioni IAM](#).

Le seguenti sezioni contengono le autorizzazioni richieste con esempi.

Per utilizzare i criteri di esempio `<REGION><AWS ACCOUNT ID>`, sostituisci e stringhe simili con i valori appropriati.

Le seguenti politiche di esempio includono Amazon Resource Names (ARNs) per le risorse. Se state lavorando nelle partizioni AWS GovCloud (US) o in AWS Cina, queste ARNs devono essere cambiate. In particolare, devono essere modificati da «arn:aws» a «arn:aws-us-gov» per la AWS GovCloud (US) partizione o «arn:aws-cn» per la partizione cinese. AWS Per ulteriori informazioni, consulta [Amazon Resource Names \(ARNs\) in AWS GovCloud \(US\) Regions](#) nella AWS GovCloud (US) User Guide e [ARNs per AWS i servizi in Cina in](#) Getting Started with AWS services in China.

Puoi tenere traccia delle modifiche alle politiche di esempio nella [AWS ParallelCluster documentazione su GitHub](#).

Argomenti

- [AWS ParallelCluster Ruoli delle istanze Amazon EC2](#)
- [AWS ParallelCluster esempi di politiche pcluster utente](#)
- [AWS ParallelCluster esempi di politiche utente per la gestione delle risorse IAM](#)
- [AWS ParallelCluster parametri di configurazione per gestire le autorizzazioni IAM](#)

AWS ParallelCluster Ruoli delle istanze Amazon EC2

Quando crei un cluster con le impostazioni di configurazione predefinite, AWS ParallelCluster utilizza i [profili di istanza](#) Amazon EC2 per creare automaticamente un ruolo di [istanza Amazon EC2](#) del cluster predefinito che fornisce le autorizzazioni necessarie per creare e gestire il cluster e le relative risorse.

Alternative all'utilizzo del ruolo di istanza predefinito AWS ParallelCluster

Al posto del ruolo di AWS ParallelCluster istanza predefinito, puoi utilizzare l'impostazione di configurazione del `InstanceRole` cluster per specificare il tuo ruolo IAM esistente per EC2. Per ulteriori informazioni, consulta [AWS ParallelCluster parametri di configurazione per gestire le autorizzazioni IAM](#). In genere, si specificano i ruoli IAM esistenti per controllare completamente le autorizzazioni concesse a EC2.

Se il tuo intento è aggiungere politiche aggiuntive al ruolo di istanza predefinito, ti consigliamo di passare le politiche IAM aggiuntive utilizzando l'impostazione di `AdditionalIamPoliciesInstanceProfile` configurazione anziché le impostazioni o. `InstanceRole` Puoi eseguire l'aggiornamento `AdditionalIamPolicies` quando aggiorni il cluster, tuttavia non puoi aggiornare `InstanceRole` quando aggiorni il cluster.

AWS ParallelCluster esempi di politiche **pcluster** utente

Gli esempi seguenti mostrano le politiche utente necessarie per creare e gestire AWS ParallelCluster e le relative risorse utilizzando la `pcluster` CLI. È possibile allegare politiche a un utente o a un ruolo.

Argomenti

- [Politica AWS ParallelCluster `pcluster` utente di base](#)
- [Politica AWS ParallelCluster `pcluster` utente aggiuntiva quando si utilizza lo scheduler AWS Batch](#)
- [Politica AWS ParallelCluster `pcluster` utente aggiuntiva per l'utilizzo di Amazon FSx for Lustre](#)
- [AWS ParallelCluster policy `pcluster` utente di image build](#)

Politica AWS ParallelCluster **pcluster** utente di base

La seguente politica mostra le autorizzazioni necessarie per eseguire AWS ParallelCluster `pcluster` i comandi.

L'ultima azione elencata nella policy è inclusa per fornire la convalida di tutti i segreti specificati nella configurazione del cluster. Ad esempio, un Gestione dei segreti AWS segreto viene utilizzato per configurare l'[DirectoryService](#) integrazione. In questo caso, un cluster viene creato solo se esiste un segreto valido in [PasswordSecretArn](#). Se questa azione viene omessa, la convalida segreta viene ignorata. Per migliorare il livello di sicurezza, si consiglia di circoscrivere questa informativa aggiungendo solo i segreti specificati nella configurazione del cluster.

Note

Se i file system Amazon EFS esistenti sono gli unici file system utilizzati nel cluster, puoi estendere le istruzioni politiche di esempio di Amazon EFS ai file system specifici a cui si fa riferimento nel file [Sezione SharedStorage](#) di configurazione del cluster.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Read"
    },
    {
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateFleet",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateNetworkInterface",
        "ec2:CreatePlacementGroup",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateTags",
        "ec2>DeleteTags",
        "ec2:CreateVolume",
        "ec2>DeleteLaunchTemplate",
        "ec2>DeleteNetworkInterface",
        "ec2>DeletePlacementGroup",
        "ec2>DeleteSecurityGroup",
        "ec2>DeleteVolume",
        "ec2:DisassociateAddress",
        "ec2:ModifyLaunchTemplate",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifyVolume",
        "ec2:ModifyVolumeAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
```

```

        "ec2:RunInstances",
        "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Write"
},
{
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTagsOfResource",
        "dynamodb:CreateTable",
        "dynamodb>DeleteTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:Query",
        "dynamodb:TagResource",
        "dynamodb:UntagResource"
    ],
    "Resource": "arn:aws:dynamodb:*:111122223333:table/parallelcluster-
*",
    "Effect": "Allow",
    "Sid": "DynamoDB"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:CreateHostedZone",
        "route53>DeleteHostedZone",
        "route53:GetChange",
        "route53:GetHostedZone",
        "route53:ListResourceRecordSets",
        "route53:ListQueryLoggingConfigs"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
},
{
    "Action": [
        "cloudformation:*"
    ],

```



```

        "Resource": "*",
        "Effect": "Allow",
        "Sid": "CloudFormation"
    },
    {
        "Action": [
            "cloudwatch:PutDashboard",
            "cloudwatch:ListDashboards",
            "cloudwatch:DeleteDashboards",
            "cloudwatch:GetDashboard",
            "cloudwatch:PutMetricAlarm",
            "cloudwatch:DeleteAlarms",
            "cloudwatch:DescribeAlarms",
            "cloudwatch:PutCompositeAlarm",
            "cloudwatch:TagResource",
            "cloudwatch:UntagResource"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "CloudWatch"
    },
    {
        "Action": [
            "iam:GetRole",
            "iam:GetRolePolicy",
            "iam:GetPolicy",
            "iam:SimulatePrincipalPolicy",
            "iam:GetInstanceProfile"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:role/*",
            "arn:aws:iam::111122223333:policy/*",
            "arn:aws:iam::aws:policy/*",
            "arn:aws:iam::111122223333:instance-profile/*"
        ],
        "Effect": "Allow",
        "Sid": "IamRead"
    },
    {
        "Action": [
            "iam:CreateInstanceProfile",
            "iam>DeleteInstanceProfile",
            "iam:AddRoleToInstanceProfile",
            "iam:RemoveRoleFromInstanceProfile"
        ]
    }

```

```

    ],
    "Resource": [
        "arn:aws:iam::111122223333:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamInstanceProfile"
},
{
    "Condition": {
        "StringEqualsIfExists": {
            "iam:PassedToService": [
                "lambda.amazonaws.com",
                "ec2.amazonaws.com",
                "spotfleet.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda:DeleteFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:GetFunction",
        "lambda:InvokeFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:UpdateFunctionConfiguration",
        "lambda:TagResource",
        "lambda:ListTags",
        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:aws:lambda:*:111122223333:function:parallelcluster-*",
        "arn:aws:lambda:*:111122223333:function:pcluster-*"
    ],

```

```

        "Effect": "Allow",
        "Sid": "Lambda"
    },
    {
        "Action": [
            "s3:*"
        ],
        "Resource": [
            "arn:aws:s3::parallelcluster-*",
            "arn:aws:s3::aws-parallelcluster-*"
        ],
        "Effect": "Allow",
        "Sid": "S3ResourcesBucket"
    },
    {
        "Action": [
            "s3:Get*",
            "s3:List*"
        ],
        "Resource": "arn:aws:s3::*-aws-parallelcluster*",
        "Effect": "Allow",
        "Sid": "S3ParallelClusterReadOnly"
    },
    {
        "Action": [
            "elasticfilesystem:*"
        ],
        "Resource": [
            "arn:aws:elasticfilesystem:*:111122223333:*"
        ],
        "Effect": "Allow",
        "Sid": "EFS"
    },
    {
        "Action": [
            "logs:DeleteLogGroup",
            "logs:PutRetentionPolicy",
            "logs:DescribeLogGroups",
            "logs:CreateLogGroup",
            "logs:TagResource",
            "logs:UntagResource",
            "logs:FilterLogEvents",
            "logs:GetLogEvents",
            "logs:CreateExportTask",

```

```

        "logs:DescribeLogStreams",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:PutMetricFilter",
        "logs:DeleteMetricFilter",
        "logs:ListTagsForResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogs"
},
{
    "Action": [
        "resource-groups:ListGroupResources"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ResourceGroupRead"
},
{
    "Sid": "AllowDescribingFileCache",
    "Effect": "Allow",
    "Action": [
        "fsx:DescribeFileCaches"
    ],
    "Resource": "*"
},
{
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "arn:aws:secretsmanager:us-  
east-1:111122223333:secret:<SECRET NAME>",
    "Effect": "Allow"
}
]
}

```

Politica AWS ParallelCluster **pcluster** utente aggiuntiva quando si utilizza lo scheduler AWS Batch

Nel caso in cui sia necessario creare e gestire un cluster con AWS Batch scheduler, è richiesta la seguente politica aggiuntiva.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEqualsIfExists": {
          "iam:PassedToService": [
            "ecs-tasks.amazonaws.com",
            "batch.amazonaws.com",
            "codebuild.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamPassRole"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/aws-service-role/batch.amazonaws.com/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

```

        "Action": [
            "codebuild:*"
        ],
        "Resource": "arn:aws:codebuild:*:111122223333:project/pcluster-*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "ecr:*"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "ECR"
    },
    {
        "Action": [
            "batch:*"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "Batch"
    },
    {
        "Action": [
            "events:*"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "AmazonCloudWatchEvents"
    },
    {
        "Action": [
            "ecs:DescribeContainerInstances",
            "ecs:ListContainerInstances"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "ECS"
    }
]
}

```

Politica AWS ParallelCluster **pcluster** utente aggiuntiva per l'utilizzo di Amazon FSx for Lustre

Nel caso in cui sia necessario creare e gestire un cluster con Amazon FSx for Lustre, è richiesta la seguente politica aggiuntiva.

Note

Se i FSx file system Amazon esistenti sono gli unici file system utilizzati nel tuo cluster, puoi limitare le dichiarazioni di FSx policy di Amazon di esempio ai file system specifici a cui si fa riferimento nel file [Sezione SharedStorage](#) di configurazione del cluster.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "fsx.amazonaws.com",
            "s3.data-source.lustre.fsx.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "fsx:*"
      ],
      "Resource": [
        "arn:aws:fsx:*:111122223333:*"
      ],
    }
  ]
}
```

```

        "Effect": "Allow",
        "Sid": "FSx"
    },
    {
        "Action": [
            "iam:CreateServiceLinkedRole",
            "iam:AttachRolePolicy",
            "iam:PutRolePolicy"
        ],
        "Resource": "arn:aws:iam::111122223333:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:Get*",
            "s3:List*",
            "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
        "Effect": "Allow"
    }
]
}

```

AWS ParallelCluster policy **pcluster** utente di image build

Gli utenti che intendono creare immagini Amazon EC2 personalizzate con AWS ParallelCluster devono disporre del seguente set di autorizzazioni.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeInstanceTypes",

```



```

        "ec2:DeregisterImage",
        "ec2:DeleteSnapshot"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2"
},
{
    "Action": [
        "iam:CreateInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:CreateRole",
        "iam:TagRole",
        "iam:GetRole",
        "iam:PutRolePolicy",
        "iam:GetRolePolicy",
        "iam:GetInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::111122223333:instance-profile/parallelcluster/*",
        "arn:aws:iam::111122223333:instance-profile/
ParallelClusterImage*",
        "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IAM"
},
{
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "lambda.amazonaws.com",
                "ec2.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::111122223333:instance-profile/parallelcluster/*",
        "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],

```

```

        "Effect": "Allow",
        "Sid": "IAMPassRole"
    },
    {
        "Action": [
            "logs:GetLogEvents",
            "logs:CreateLogGroup",
            "logs:TagResource",
            "logs:UntagResource",
            "logs>DeleteLogGroup"
        ],
        "Resource": [
            "arn:aws:logs:*:111122223333:log-group:/aws/imagebuilder/ParallelClusterImage-*",
            "arn:aws:logs:*:111122223333:log-group:/aws/lambda/ParallelClusterImage-*"
        ],
        "Effect": "Allow",
        "Sid": "CloudWatch"
    },
    {
        "Action": [
            "cloudformation:DescribeStacks",
            "cloudformation:CreateStack",
            "cloudformation>DeleteStack"
        ],
        "Resource": [
            "arn:aws:cloudformation:*:111122223333:stack/*"
        ],
        "Effect": "Allow",
        "Sid": "CloudFormation"
    },
    {
        "Action": [
            "lambda:CreateFunction",
            "lambda:GetFunction",
            "lambda:AddPermission",
            "lambda:RemovePermission",
            "lambda>DeleteFunction",
            "lambda:TagResource",
            "lambda:ListTags",
            "lambda:UntagResource"
        ],
        "Resource": [

```

```

        "arn:aws:lambda:*:111122223333:function:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
  },
  {
    "Action": [
      "imagebuilder:Get*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ImageBuilderGet"
  },
  {
    "Action": [
      "imagebuilder:CreateImage",
      "imagebuilder:TagResource",
      "imagebuilder:CreateImageRecipe",
      "imagebuilder:CreateComponent",
      "imagebuilder:CreateDistributionConfiguration",
      "imagebuilder:CreateInfrastructureConfiguration",
      "imagebuilder>DeleteImage",
      "imagebuilder>DeleteComponent",
      "imagebuilder>DeleteImageRecipe",
      "imagebuilder>DeleteInfrastructureConfiguration",
      "imagebuilder>DeleteDistributionConfiguration"
    ],
    "Resource": [
      "arn:aws:imagebuilder:*:111122223333:image/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:111122223333:image-recipe/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:111122223333:component/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:111122223333:distribution-configuration/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:111122223333:infrastructure-configuration/parallelclusterimage-*"
    ],
    "Effect": "Allow",
    "Sid": "ImageBuilder"
  },
  {
    "Action": [

```

```

        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:ListBucketVersions"
    ],
    "Resource": [
        "arn:aws:s3:::parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3Bucket"
},
{
    "Action": [
        "sns:GetTopicAttributes",
        "sns:TagResource",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Publish",
        "SNS:DeleteTopic",
        "SNS:Unsubscribe"
    ],
    "Resource": [
        "arn:aws:sns:*:111122223333:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "SNS"
},
{
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
    ],
    "Resource": [
        "arn:aws:s3:::parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "S3Objects"
},
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",

```

```
    "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "imagebuilder.amazonaws.com"
        }
    }
}
```

AWS ParallelCluster esempi di politiche utente per la gestione delle risorse IAM

Quando vengono utilizzate AWS ParallelCluster per creare cluster o personalizzate AMIs, è necessario fornire politiche IAM che contengano le autorizzazioni per concedere il set di autorizzazioni richiesto ai componenti. AWS ParallelCluster Queste risorse IAM possono essere create automaticamente AWS ParallelCluster o fornite come input durante la creazione di un cluster o di un'immagine personalizzata.

Puoi utilizzare le seguenti modalità per fornire all' AWS ParallelCluster utente le autorizzazioni necessarie per accedere alle risorse IAM utilizzando politiche IAM aggiuntive nella configurazione.

Argomenti

- [Modalità di accesso IAM privilegiata](#)
- [Modalità di accesso IAM con restrizioni](#)
- [Modalità PermissionsBoundary](#)

Modalità di accesso IAM privilegiata

Con questa modalità, crea AWS ParallelCluster automaticamente tutte le risorse IAM necessarie. Queste policy IAM sono ridotte in modo da consentire l'accesso solo alle risorse del cluster.

Per abilitare la modalità di accesso IAM privilegiata, aggiungi la seguente policy al ruolo utente.

 Note

Se [AdditionalPolicies](#) configuri i parametri [HeadNodeIam//AdditionalPolicies](#) o [SchedulingSlurmQueues//Iam](#), devi fornire AWS ParallelCluster all'utente l'autorizzazione ad allegare e scollegare le politiche di ruolo per ogni politica aggiuntiva, come illustrato nella seguente politica. Aggiungi la politica aggiuntiva alla condizione ARNs per allegare e scollegare le politiche relative ai ruoli.

 Warning

Questa modalità consente all'utente di disporre dei privilegi di amministratore IAM nel Account AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:DeleteRole",
        "iam:TagRole",
        "iam:UntagRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRole"
    },
    {
      "Action": [
        "iam:CreateRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
      ],
    }
  ]
}
```

```

        "Effect": "Allow",
        "Sid": "IamCreateRole"
    },
    {
        "Action": [
            "iam:PutRolePolicy",
            "iam>DeleteRolePolicy"
        ],
        "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
        "Effect": "Allow",
        "Sid": "IamInlinePolicy"
    },
    {
        "Condition": {
            "ArnLike": {
                "iam:PolicyARN": [
                    "arn:aws:iam::111122223333:policy/parallelcluster*",
                    "arn:aws:iam::111122223333:policy/parallelcluster/*",
                    "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
                    "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
                    "arn:aws:iam::aws:policy/AWSBatchFullAccess",
                    "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
                    "arn:aws:iam::aws:policy/service-role/
AWSBatchServiceRole",
                    "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
                    "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
                    "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
                    "arn:aws:iam::aws:policy/
EC2InstanceProfileForImageBuilder",
                    "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
                ]
            }
        },
        "Action": [
            "iam:AttachRolePolicy",
            "iam:DetachRolePolicy"
        ],
        "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
        "Effect": "Allow",
        "Sid": "IamPolicy"
    }

```

```
    }
  ]
}
```

Modalità di accesso IAM con restrizioni

Quando all'utente non vengono concesse politiche IAM aggiuntive, i ruoli IAM richiesti dai cluster o dalla creazione di immagini personalizzate devono essere creati manualmente da un amministratore e passati come parte della configurazione del cluster.

Quando si crea un cluster sono richiesti i seguenti parametri:

- [Iam](#) / [Roles](#) / [LambdaFunctionsRole](#)
- [HeadNode](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)
- [Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Quando si crea un'immagine personalizzata, sono richiesti i seguenti parametri:

- [Build](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)
- [Build](#) / [Iam](#) / [CleanupLambdaRole](#)

I ruoli IAM passati come parte dei parametri sopra elencati devono essere creati nel prefisso del `/parallelcluster/` percorso. Se ciò non è possibile, la politica dell'utente deve essere aggiornata per concedere l'`iam:PassRole` autorizzazione su ruoli personalizzati specifici, come nell'esempio seguente.

```
{
  "Condition": {
    "StringEqualsIfExists": {
      "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "lambda.amazonaws.com",
        "ec2.amazonaws.com",
        "spotfleet.amazonaws.com",
        "batch.amazonaws.com",
        "codebuild.amazonaws.com"
      ]
    }
  },
}
```



```
"Action": [
    "iam:PassRole"
],
"Resource": [
    <list all custom IAM roles>
],
"Effect": "Allow",
"Sid": "IamPassRole"
}
```

Warning

Attualmente questa modalità non consente la gestione dei AWS Batch cluster perché non tutti i ruoli IAM possono essere passati nella configurazione del cluster.

Modalità **PermissionsBoundary**

Questa modalità delega AWS ParallelCluster alla creazione di ruoli IAM associati al limite delle autorizzazioni IAM configurate. Per ulteriori informazioni sui limiti delle autorizzazioni IAM, consulta Limiti delle [autorizzazioni per le entità IAM nella Guida per l'utente IAM](#).

La seguente policy deve essere aggiunta al ruolo utente.

Nella policy, sostituiscilo *<permissions-boundary-arn>* con l'ARN della policy IAM da applicare come limite delle autorizzazioni.

Warning

Se configuri i [AdditionalPolicies](#) parametri [HeadNode/Iam/AdditionalPolicies](#) o [Scheduling/SlurmQueuesIam](#)/, devi concedere all'utente l'autorizzazione ad allegare e scollegare le politiche di ruolo per ogni politica aggiuntiva, come illustrato nella politica seguente. Aggiungi la politica aggiuntiva alla condizione ARNs per allegare e scollegare le politiche relative ai ruoli.

JSON

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

```

"Statement": [
  {
    "Action": [
      "iam:CreateServiceLinkedRole",
      "iam:DeleteRole",
      "iam:TagRole"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          "<permissions-boundary-arn>"
        ]
      }
    },
    "Action": [
      "iam:CreateRole"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamCreateRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          "<permissions-boundary-arn>"
        ]
      }
    },
    "Action": [
      "iam:PutRolePolicy",
      "iam:DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
    "Effect": "Allow",
  }
]

```

```

        "Sid": "IamInlinePolicy"
    },
    {
        "Condition": {
            "StringEquals": {
                "iam:PermissionsBoundary": [
                    "<permissions-boundary-arn>"
                ]
            },
            "ArnLike": {
                "iam:PolicyARN": [
                    "arn:aws:iam::111122223333:policy/parallelcluster*",
                    "arn:aws:iam::111122223333:policy/parallelcluster/*",
                    "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
                    "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
                    "arn:aws:iam::aws:policy/AWSBatchFullAccess",
                    "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
                    "arn:aws:iam::aws:policy/service-role/
AWSBatchServiceRole",
                    "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
                    "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
                    "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
                    "arn:aws:iam::aws:policy/
EC2InstanceProfileForImageBuilder",
                    "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
                ]
            }
        },
        "Action": [
            "iam:AttachRolePolicy",
            "iam:DetachRolePolicy"
        ],
        "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
        "Effect": "Allow",
        "Sid": "IamPolicy"
    }
]
}

```

Quando questa modalità è abilitata, è necessario specificare il limite delle autorizzazioni ARN nel parametro di [PermissionsBoundary](#) configurazione [Iam](#)/durante la creazione o l'aggiornamento di un cluster e nel [PermissionBoundary](#) parametro [Build/Iam](#)/quando si crea un'immagine personalizzata.

AWS ParallelCluster parametri di configurazione per gestire le autorizzazioni IAM

AWS ParallelCluster espone una serie di opzioni di configurazione per personalizzare e gestire le autorizzazioni e i ruoli IAM utilizzati in un cluster o durante il processo di creazione di AMI personalizzate.

Argomenti

- [Configurazione del cluster](#)
- [Configurazione personalizzata dell'immagine](#)

Configurazione del cluster

Argomenti

- [Ruolo IAM del nodo principale](#)
- [Accesso ad Amazon S3](#)
- [Politiche IAM aggiuntive](#)
- [AWS Lambda funzioni \(ruolo\)](#)
- [Nodi di calcolo \(ruolo IAM\)](#)
- [Limite delle autorizzazioni](#)

Ruolo IAM del nodo principale

[HeadNode](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Con questa opzione, sostituisci il ruolo IAM predefinito assegnato al nodo principale del cluster. Per ulteriori dettagli, consulta il [InstanceProfile](#) riferimento.

Ecco il set minimo di politiche da utilizzare come parte di questo ruolo quando lo scheduler è Slurm:

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` politica IAM gestita. Per ulteriori informazioni, consulta [Creare ruoli e utenti IAM da utilizzare con l' CloudWatch agente](#) nella Amazon CloudWatch User Guide.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politica IAM gestita. Per ulteriori informazioni, consulta la sezione [AWS dedicata alle politiche gestite AWS Systems Manager nella Guida per AWS Systems Manager](#) l'utente.
- Policy IAM aggiuntiva:
JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::us-east-1-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.us-east-1/*",
        "arn:aws:s3:::parallelcluster-*v1-do-not-delete/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:BatchGetItem"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:111122223333:table/parallelcluster-*",
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/parallelcluster:node-type": "Compute"
        }
      }
    }
  ]
}
```

```

        "Action": "ec2:TerminateInstances",
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "ec2:RunInstances",
            "ec2:CreateFleet"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": [
                    "ec2.amazonaws.com"
                ]
            }
        },
        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:role/parallelcluster/*",
            "arn:aws:iam::111122223333:instance-profile/parallelcluster/*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "ec2:DescribeInstances",
            "ec2:DescribeInstanceStatus",
            "ec2:DescribeVolumes",
            "ec2:DescribeInstanceAttribute",
            "ec2:DescribeCapacityReservations"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "ec2:CreateTags",
            "ec2:AttachVolume"

```

```

    ],
    "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackResource",
        "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:<SECRET_ID>",
    "Effect": "Allow"
}
]
}

```

Nota che nel caso in cui [SchedulingSlurmQueues/Iam//InstanceRole](#) venga utilizzato per sovrascrivere il ruolo IAM di calcolo, la policy del nodo principale riportata sopra deve includere tale ruolo nella Resource sezione dell'iam:PassRole autorizzazione.

Ecco il set minimo di politiche da utilizzare come parte di questo ruolo quando lo scheduler è: AWS Batch

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` politica IAM gestita. Per ulteriori informazioni, consulta [Creare ruoli e utenti IAM da utilizzare con l' CloudWatch agente](#) nella Amazon CloudWatch User Guide.

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politica IAM gestita. Per ulteriori informazioni, consulta la sezione [AWS dedicata alle politiche gestite AWS Systems Manager nella Guida per AWS Systems Manager](#) l'utente.
- Policy IAM aggiuntiva:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::dcv-license.us-east-1/*",
        "arn:aws:s3:::us-east-1-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*",
        "arn:aws:iam::111122223333:instance-profile/parallelcluster/*"
      ]
    }
  ]
}
```



```

    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "batch:DescribeJobQueues",
      "batch:DescribeJobs",
      "batch:ListJobs",
      "batch:DescribeComputeEnvironments"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "batch:SubmitJob",
      "batch:TerminateJob",
      "logs:GetLogEvents",
      "ecs:ListContainerInstances",
      "ecs:DescribeContainerInstances"
    ],
    "Resource": [
      "arn:aws:logs:us-east-1:111122223333:log-group:/aws/batch/
      job:log-stream:PclusterJobDefinition*",
      "arn:aws:ecs:us-east-1:111122223333:container-instance/
      AWSBatch-PclusterComputeEnviron*",
      "arn:aws:ecs:us-east-1:111122223333:cluster/AWSBatch-
      Pcluster*",
      "arn:aws:batch:us-east-1:111122223333:job-queue/
      PclusterJobQueue*",
      "arn:aws:batch:us-east-1:111122223333:job-definition/
      PclusterJobDefinition*:*",
      "arn:aws:batch:us-east-1:111122223333:job/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeVolumes",
      "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*",

```

```

        "Effect": "Allow"
    },
    {
        "Action": [
            "ec2:CreateTags",
            "ec2:AttachVolume"
        ],
        "Resource": [
            "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "arn:aws:ec2:us-east-1:111122223333:volume/*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "cloudformation:DescribeStackResource",
            "cloudformation:DescribeStacks",
            "cloudformation:SignalResource"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Action": "secretsmanager:GetSecretValue",
        "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:<SECRET_ID>",
        "Effect": "Allow"
    }
]
}

```

Accesso ad Amazon S3

[HeadNode/Iam/S3Accesso](#) [Scheduling/SlurmQueuesS3Access](#)

In queste sezioni di configurazione, puoi personalizzare l'accesso ad Amazon S3 concedendo policy Amazon S3 aggiuntive ai ruoli IAM associati al nodo principale o ai nodi di calcolo del cluster quando tali ruoli vengono creati da AWS ParallelCluster. Per ulteriori informazioni, consulta la documentazione di riferimento per ogni parametro di configurazione.

Questo parametro può essere utilizzato solo quando l'utente è configurato con [Modalità di accesso IAM privilegiata](#) o [Modalità PermissionsBoundary](#).

Politiche IAM aggiuntive

[HeadNode/Iam/AdditionalIamPolicies](#) o [SlurmQueuesIam/AdditionalIamPolicies](#)

Utilizza questa opzione per allegare politiche IAM gestite aggiuntive ai ruoli IAM associati al nodo principale o ai nodi di calcolo del cluster quando tali ruoli vengono creati da AWS ParallelCluster.

Warning

Per utilizzare questa opzione, assicurati che all'[AWS ParallelCluster utente](#) siano `iam:AttachRolePolicy` concesse `iam:DetachRolePolicy` le autorizzazioni per le policy IAM che devono essere allegate.

AWS Lambda funzioni (ruolo)

[Iam](#) / [Roles](#) / [LambdaFunctionsRole](#)

Questa opzione sovrascrive il ruolo associato a tutte le AWS Lambda funzioni utilizzate durante il processo di creazione del cluster. AWS Lambda deve essere configurato come principale autorizzato ad assumere il ruolo.

Note

Se [DeploymentSettings/LambdaFunctionsVpcConfig](#) è impostato, `LambdaFunctionsRole` deve includere l'[autorizzazione del AWS Lambda ruolo](#) per impostare la configurazione del VPC.

Ecco il set minimo di politiche da utilizzare come parte di questo ruolo:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],

```

```

        "Resource": "arn:aws:route53::hostedzone/*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "logs:CreateLogStream",
            "logs:PutLogEvents"
        ],
        "Effect": "Allow",
        "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
lambda/pcluster-*"
    },
    {
        "Action": "ec2:DescribeInstances",
        "Effect": "Allow",
        "Resource": "*"
    },
    {
        "Action": "ec2:TerminateInstances",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/parallelcluster:node-type": "Compute"
            }
        },
        "Effect": "Allow",
        "Resource": "*"
    },
    {
        "Action": [
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:ListBucket",
            "s3:ListBucketVersions"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3::parallelcluster-*-v1-do-not-delete",
            "arn:aws:s3::parallelcluster-*-v1-do-not-delete/*"
        ]
    }
]
}

```

Nodi di calcolo (ruolo IAM)

[Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Questa opzione consente di sovrascrivere il ruolo IAM assegnato ai nodi di calcolo del cluster. Per ulteriori informazioni, consulta [InstanceProfile](#).

Ecco il set minimo di policy da utilizzare come parte di questo ruolo:

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` politica IAM gestita. Per ulteriori informazioni, consulta [Creare ruoli e utenti IAM da utilizzare con l' CloudWatchagente](#) nella Amazon CloudWatch User Guide.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politica IAM gestita. Per ulteriori informazioni, consulta la sezione [AWS dedicata alle politiche gestite AWS Systems Manager nella Guida per AWS Systems Manager](#) l'utente.
- Policy IAM aggiuntiva:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:Query",
        "dynamodb:UpdateItem",
        "dynamodb:PutItem",
        "dynamodb:GetItem"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:111122223333:table/parallelcluster-*",
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::us-east-1-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "ec2:DescribeInstanceAttribute",
      "Resource": "*",
```

```
        "Effect": "Allow"
      },
      {
        "Action": "cloudformation:DescribeStackResource",
        "Resource": [
          "arn:aws:cloudformation:us-east-1:111122223333:stack/*/*"
        ],
        "Effect": "Allow"
      }
    ]
  }
}
```

Limite delle autorizzazioni

[Iam](#) / [PermissionsBoundary](#)

Questo parametro impone AWS ParallelCluster di associare la policy IAM specificata come PermissionsBoundary a tutti i ruoli IAM creati come parte di una distribuzione del cluster.

Vedi [Modalità PermissionsBoundary](#) l'elenco delle politiche richieste dall'utente quando viene definita questa impostazione.

Configurazione personalizzata dell'immagine

Argomenti

- [Ruolo di istanza per EC2 Image Builder](#)
- [AWS Lambda ruolo di pulizia](#)
- [Politiche IAM aggiuntive](#)
- [Limite delle autorizzazioni](#)

Ruolo di istanza per EC2 Image Builder

[Build](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Con questa opzione sovrascrivi il ruolo IAM assegnato all'istanza Amazon EC2 lanciata da EC2 Image Builder per creare un'AMI personalizzata.

Ecco il set minimo di policy da utilizzare come parte di questo ruolo:

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCorepolicy` IAM gestita. Per ulteriori informazioni, consulta la sezione [AWS dedicata alle politiche gestite AWS Systems Manager nella Guida per AWS Systems Manager](#) l'utente.
- `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilderpolicy` IAM gestita. Per ulteriori informazioni, vedere la [EC2InstanceProfileForImageBuilderpolicy nella Guida](#) per l'utente di Image Builder.
- Policy IAM aggiuntiva:
JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateTags",
        "ec2:ModifyImageAttribute"
      ],
      "Resource": "arn:aws:ec2:us-east-1:image/*",
      "Effect": "Allow"
    }
  ]
}
```

AWS Lambda ruolo di pulizia

[Build](#) / [Iam](#) / [CleanupLambdaRole](#)

Questa opzione sostituisce il ruolo associato a tutte le AWS Lambda funzioni utilizzate durante il processo di creazione dell'immagine personalizzata. AWS Lambda deve essere configurato come principale autorizzato ad assumere il ruolo.

 Note

Se [DeploymentSettings/LambdaFunctionsVpcConfig](#) è impostato, `CleanupLambdaRole` deve includere l'[autorizzazione del AWS Lambda ruolo](#) per impostare la configurazione del VPC.

Ecco il set minimo di politiche da utilizzare come parte di questo ruolo:

- `arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` politica IAM gestita. Per ulteriori informazioni, consulta [le politiche AWS gestite per le funzionalità Lambda nella Guida](#) per gli AWS Lambda sviluppatori.
- Policy IAM aggiuntiva:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:DetachRolePolicy",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy"
      ],
      "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam>DeleteInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
      "Resource": "arn:aws:iam::111122223333:instance-profile/parallelcluster/*",
      "Effect": "Allow"
    },
    {
      "Action": "imagebuilder:DeleteInfrastructureConfiguration",
      "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:infrastructure-configuration/parallelclusterimage-*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "imagebuilder>DeleteComponent"
      ],
      "Resource": [
        "arn:aws:imagebuilder:us-east-1:111122223333:component/parallelclusterimage-*/*"
      ]
    }
  ]
}
```



```

    ],
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteImageRecipe",
    "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:image-recipe/parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteDistributionConfiguration",
    "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:distribution-configuration/parallelclusterimage-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder:DeleteImage",
      "imagebuilder:GetImage",
      "imagebuilder:CancelImageCreation"
    ],
    "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:image/parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "cloudformation:DeleteStack",
    "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*/*",
    "Effect": "Allow"
  },
  {
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:us-east-1::image/*",
    "Effect": "Allow"
  },
  {
    "Action": "tag:TagResources",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:DeleteFunction",

```

```

        "lambda:RemovePermission"
    ],
    "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:ParallelClusterImage-*",
    "Effect": "Allow"
},
{
    "Action": "logs:DeleteLogGroup",
    "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
lambda/ParallelClusterImage-*:*",
    "Effect": "Allow"
},
{
    "Action": [
        "SNS:GetTopicAttributes",
        "SNS:DeleteTopic",
        "SNS:GetSubscriptionAttributes",
        "SNS:Unsubscribe"
    ],
    "Resource": "arn:aws:sns:us-
east-1:111122223333:ParallelClusterImage-*",
    "Effect": "Allow"
}
]
}

```

Politiche IAM aggiuntive

[Build](#) / [Iam](#) / [AdditionalIamPolicies](#)

Questa opzione viene utilizzata per allegare ulteriori policy IAM gestite al ruolo associato all'istanza Amazon EC2 utilizzata da EC2 Image Builder per produrre l'AMI personalizzata.

Warning

Per utilizzare questa opzione, assicurati che [AWS ParallelCluster all'utente](#) siano `iam:AttachRolePolicy` concesse `iam:DetachRolePolicy` le autorizzazioni per le policy IAM che devono essere allegate.

Limite delle autorizzazioni

[Build](#) / [Iam](#) / [PermissionsBoundary](#)

Questo parametro impone AWS ParallelCluster di allegare la policy IAM specificata come `PermissionsBoundary` a tutti i ruoli IAM creati come parte di una build AMI personalizzata.

Consulta [Modalità PermissionsBoundary](#) l'elenco delle politiche necessarie per utilizzare tale funzionalità.

Configurazioni di rete

AWS ParallelCluster utilizza Amazon Virtual Private Cloud (VPC) per il networking. VPC offre una piattaforma di rete flessibile e configurabile in cui è possibile implementare i cluster.

Il VPC deve avere `DNS Resolution = yes` opzioni DHCP con il nome di dominio corretto per la regione. `DNS Hostnames = yes` Il set di opzioni DHCP predefinito specifica già il DNS richiesto. `AmazonProvided` Se specifichi più di un server di nomi di dominio, consulta i [set di opzioni DHCP](#) nella Amazon VPC User Guide.

AWS ParallelCluster supporta le seguenti configurazioni di alto livello:

- Una sottorete per i nodi principali e di calcolo.
- Due sottoreti, con il nodo principale in una sottorete pubblica e nodi di calcolo in una sottorete privata. Le sottoreti possono essere nuove o esistenti.

Tutte queste configurazioni possono funzionare con o senza indirizzi IP pubblici. AWS ParallelCluster può anche essere implementato per utilizzare un proxy HTTP per tutte le AWS richieste. Le combinazioni di queste configurazioni determinano molti scenari di distribuzione. Ad esempio, è possibile configurare una singola sottorete pubblica con tutti gli accessi su Internet. In alternativa, puoi configurare una rete completamente privata utilizzando AWS Direct Connect un proxy HTTP per tutto il traffico.

A partire dalla AWS ParallelCluster versione 3.0.0 è possibile configurare `PlacementGroup` impostazioni diverse `SecurityGroups` per ogni coda. `AdditionalSecurityGroups` Per ulteriori informazioni, vedere [HeadNode/Networking](#) e [SlurmQueues/Networking](#). [AwsBatchQueuesNetworking](#)

Per illustrazioni di alcuni scenari di rete, vedere i seguenti diagrammi di architettura.

Argomenti

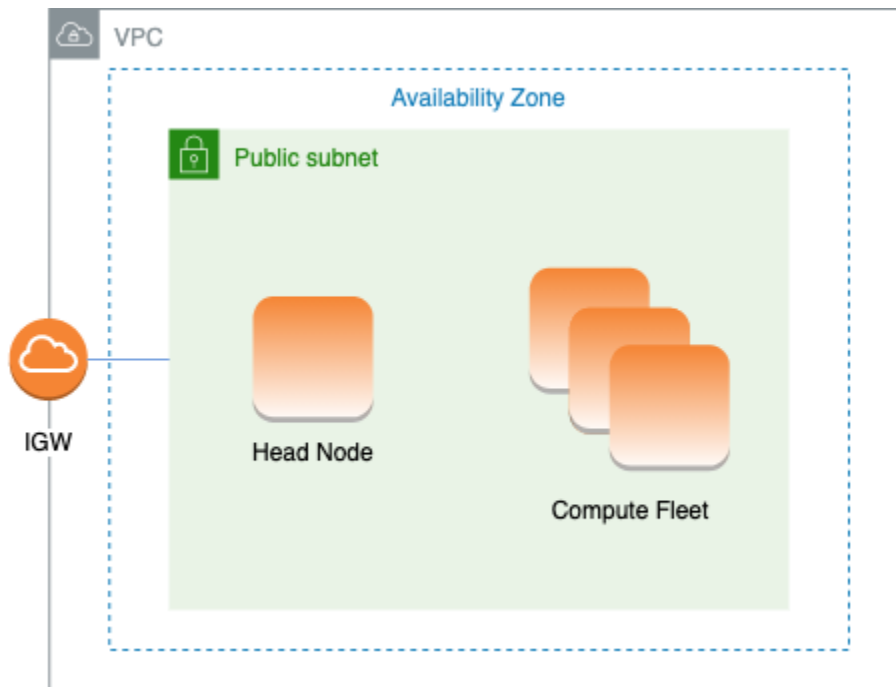
- [AWS ParallelCluster in un'unica sottorete pubblica](#)
- [AWS ParallelCluster utilizzando due sottoreti](#)
- [AWS ParallelCluster in una singola sottorete privata connessa tramite AWS Direct Connect](#)
- [AWS ParallelCluster con scheduler AWS Batch](#)
- [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#)

AWS ParallelCluster in un'unica sottorete pubblica

In questa configurazione, a tutte le istanze del cluster deve essere assegnato un IP pubblico per ottenere l'accesso a Internet. Per raggiungere questo obiettivo, effettuare le seguenti operazioni:

- Assicuratevi che al nodo principale sia assegnato un indirizzo IP pubblico attivando l'impostazione «Abilita l'assegnazione automatica dell' IPv4 indirizzo pubblico» per la sottorete utilizzata in [HeadNode/Networking/SubnetId](#) assegnando un IP elastico in [HeadNodeNetworkingElasticIp](#).
- Assicuratevi che ai nodi di calcolo sia assegnato un indirizzo IP pubblico attivando l'impostazione «Abilita l'assegnazione automatica dell' IPv4 indirizzo pubblico» per la sottorete utilizzata in [Scheduling/SlurmQueuesNetworking/SubnetId](#) impostando [AssignPublicIp](#): true in [SchedulingSlurmQueuesNetworking](#).
- Se definisci un tipo di p4d istanza o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete per il nodo principale, devi impostare [HeadNode/Networking/ElasticIp](#) per true fornire l'accesso pubblico. AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. In questo caso, si consiglia di utilizzare un [gateway NAT](#) per fornire l'accesso pubblico ai nodi di calcolo del cluster. Per ulteriori informazioni sugli indirizzi IP, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Amazon EC2 User Guide for Linux Instances.
- Non puoi definire un tipo di hp6id istanza p4d or o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete per calcolare i nodi perché AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. Per ulteriori informazioni sugli indirizzi IP, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Amazon EC2 User Guide for Linux Instances.

Per ulteriori informazioni, consulta [Enabling Internet access](#) in Amazon VPC User Guide.



La configurazione per questa architettura richiede le seguenti impostazioni:

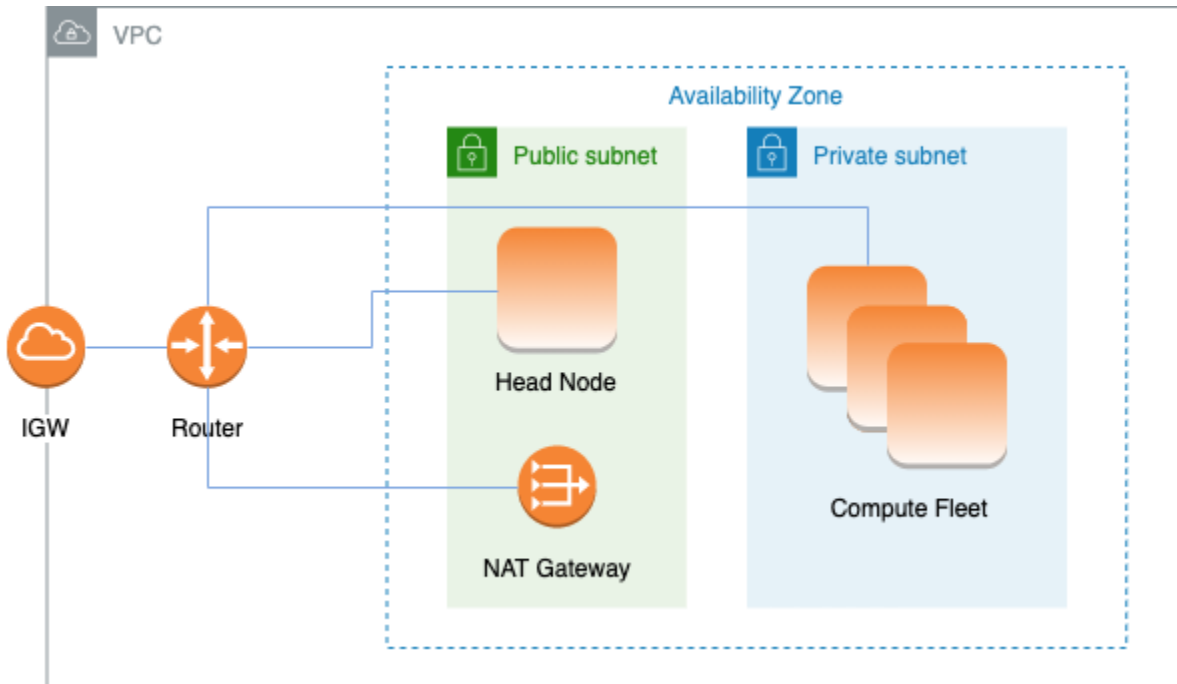
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
      Networking:
        SubnetIds:
          - subnet-12345678 # subnet with internet gateway
        #AssignPublicIp: true
```

AWS ParallelCluster utilizzando due sottoreti

In questa configurazione, solo al nodo principale del cluster è richiesto l'assegnazione di un IP pubblico. È possibile ottenere ciò attivando l'impostazione «Abilita l'assegnazione automatica dell' IPv4indirizzo pubblico» per la sottorete utilizzata in [HeadNode/Networking/SubnetId](#) assegnando un IP elastico in//. [HeadNodeNetworkingElasticIp](#)

Se definisci un tipo di istanza p4d o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete per il nodo principale, devi impostare [HeadNode//Networking](#) per fornire l'accesso pubblico. [ElasticIp](#) true AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. Per ulteriori informazioni sugli indirizzi IP, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Amazon EC2 User Guide for Linux Instances.

Questa configurazione richiede un [gateway NAT](#) o un proxy interno nella sottorete utilizzata per le code, per consentire l'accesso a Internet alle istanze di calcolo.



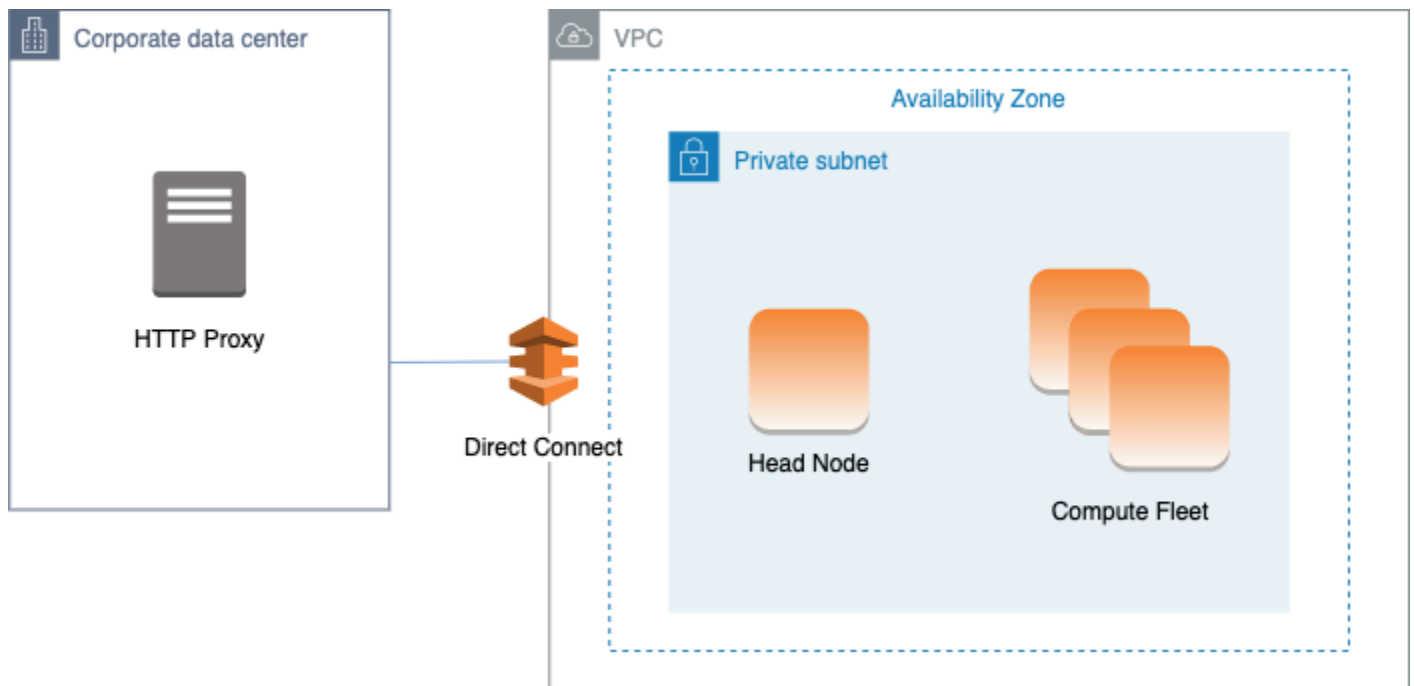
La configurazione per l'utilizzo di una sottorete privata esistente per le istanze di calcolo richiede le seguenti impostazioni:

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
```

```
- subnet-23456789 # subnet with NAT gateway
#AssignPublicIp: false
```

AWS ParallelCluster in una singola sottorete privata connessa tramite AWS Direct Connect

Quando [Scheduling/SlurmQueues/Networking/AssignPublicIp](#) è impostato su `false`, le sottoreti devono essere configurate correttamente per utilizzare il proxy per tutto il traffico. L'accesso al Web è richiesto sia per i nodi principali che per quelli di calcolo.



La configurazione per questa architettura richiede le seguenti impostazioni:

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-34567890 # subnet with proxy
    Proxy:
      HttpProxyAddress: http://proxy-address:port
  Ssh:
    KeyName: ec2-key-name
Scheduling:
  Scheduler: slurm
SlurmQueues:
```

```
- ...  
Networking:  
  SubnetIds:  
    - subnet-34567890 # subnet with proxy  
  AssignPublicIp: false  
  Proxy:  
    HttpProxyAddress: http://proxy-address:port
```

AWS ParallelCluster con scheduler AWS Batch

Quando si utilizza `awsbatch` come tipo di scheduler, AWS ParallelCluster crea un ambiente di elaborazione AWS Batch gestito. L'AWS Batch ambiente gestisce le istanze di container Amazon Elastic Container Service (Amazon ECS). Queste istanze vengono avviate nella sottorete configurata nel parametro `AwsBatchQueuesNetworkingSubnetIds`. AWS Batch Per funzionare correttamente, le istanze di container Amazon ECS necessitano dell'accesso alla rete esterna per comunicare con l'endpoint del servizio Amazon ECS. Questo si traduce negli scenari seguenti:

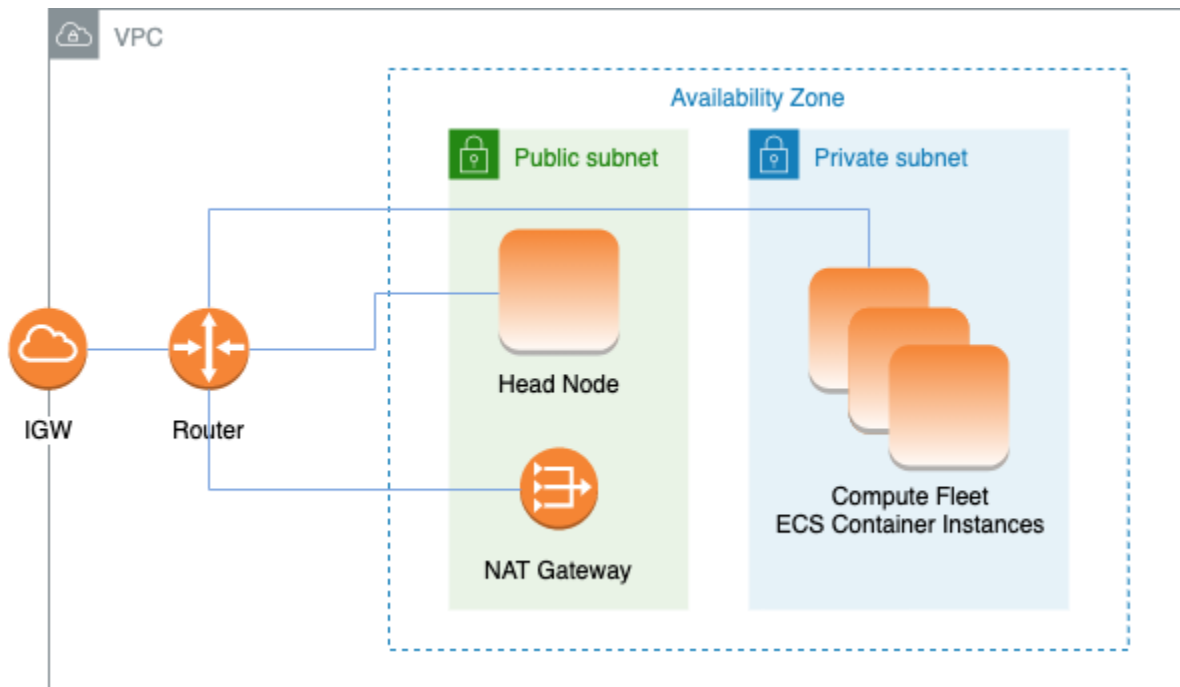
- L'ID di sottorete specificato per la coda utilizza un gateway [NAT](#) per accedere a Internet. Abbiamo consigliato questo approccio.
- Le istanze avviate nella sottorete della coda hanno indirizzi IP pubblici e possono raggiungere Internet tramite un Internet Gateway.

Inoltre, se sei interessato ai lavori paralleli multinodo (dai [AWS Batch documenti](#)):

AWS Batch i lavori paralleli a più nodi utilizzano la modalità di `awsvpc` rete Amazon ECS. Ciò conferisce ai contenitori di job 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 Amazon VPC della risorsa di elaborazione host. A questa vengono applicati anche tutti i gruppi di sicurezza applicati alle risorse di calcolo.

Quando si utilizza Amazon ECS Task Networking, la modalità di `awsvpc` rete non fornisce interfacce di rete elastiche con indirizzi IP pubblici per attività che utilizzano il tipo di avvio Amazon EC2. Per accedere a Internet, le attività che utilizzano il tipo di avvio Amazon EC2 devono essere avviate in una sottorete privata configurata per utilizzare un gateway NAT.

È necessario configurare un [gateway NAT](#) per consentire al cluster di eseguire processi paralleli multinodo.



Tutte le configurazioni e le considerazioni precedenti sono valide anche per AWS Batch. Di seguito è riportato un esempio di configurazione di AWS Batch rete.

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway, NAT gateway or proxy
    #ElasticIp: true | false | eip-12345678
    #Proxy:
      #HttpProxyAddress: http://proxy-address:port
  Ssh:
    KeyName: ec2-key-name
  Scheduling:
    Scheduler: awsbatch
  AwsBatchQueues:
    - ...
      Networking:
        SubnetIds:
          - subnet-23456789 # subnet with internet gateway, NAT gateway or proxy
          #AssignPublicIp: true | false
```

Nella [Networking](#) sezione [Scheduling/AwsBatchQueues](#), [SubnetIds](#) è un tipo di elenco ma, attualmente, è supportata solo una sottorete.

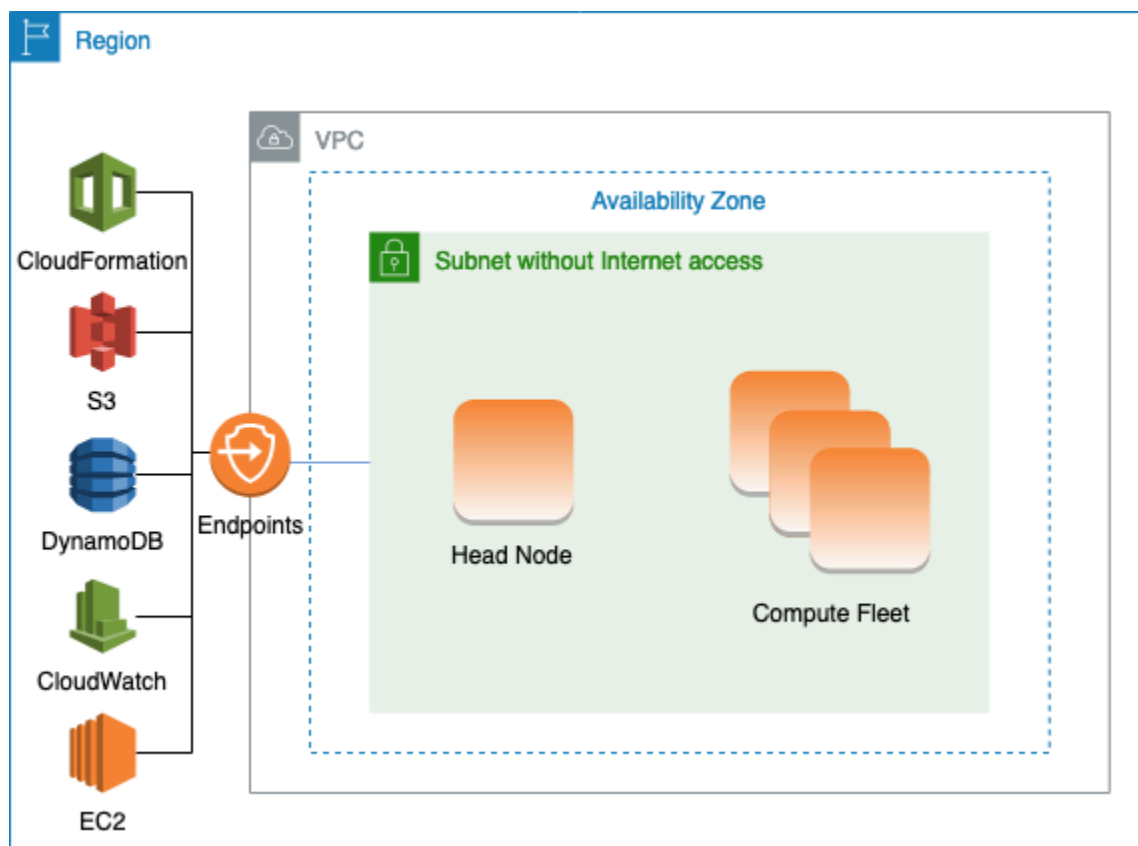
Per ulteriori informazioni, consulta i seguenti argomenti:

- [AWS Batch ambienti di elaborazione gestiti](#)
- [AWS Batch lavori paralleli multinodo](#)
- [Rete di attività di Amazon ECS con la modalità di rete awsvpc](#)

AWS ParallelCluster in un'unica sottorete senza accesso a Internet

Una sottorete senza accesso a Internet non consente connessioni in entrata o in uscita a Internet. Questa AWS ParallelCluster configurazione può aiutare i clienti interessati alla sicurezza a migliorare ulteriormente la sicurezza delle proprie risorse. AWS ParallelCluster vengono creati nodi AWS ParallelCluster AMIs che includono tutto il software necessario per eseguire un cluster senza accesso a Internet. In questo modo, AWS ParallelCluster puoi creare e gestire cluster con nodi che non dispongono di accesso a Internet.

In questa sezione, imparerai come configurare il cluster. Scopri anche le limitazioni nell'esecuzione di cluster senza accesso a Internet.



Configurazione degli endpoint VPC

Per garantire il corretto funzionamento del cluster, i nodi del cluster devono essere in grado di interagire con una serie di servizi. AWS

Crea e configura i seguenti [endpoint VPC](#) in modo che i nodi del cluster possano interagire con i AWS Servizi, senza accesso a Internet:

Commercial and AWS GovCloud (US) partitions

Servizio	Nome del servizio	Tipo
Amazon CloudWatch	com.amazonaws. <i>region-id</i> .registri	Interfaccia
CloudFormation	com.amazonaws.it. <i>region-id</i> . formazione di nuvole	Interfaccia
Amazon EC2	com.amazonaws. <i>region-id</i> .ec2	Interfaccia
Simple Storage Service (Amazon S3)	com.amazonaws.it. <i>region-id</i> .s3	Gateway
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	Gateway
Gestione dei segreti AWS**	com.amazonaws. <i>region-id</i> . gestore dei segreti	Interfaccia
AWS Elastic Load Balancing ***	com.amazonaws. <i>region-id</i> . bilanciamento elastico del carico	Interfaccia
AWS Scalabilità automatica ***	com.amazonaws. <i>region-id</i> .scalabilità automatica	Interfaccia

China partition

Servizio	Nome del servizio	Tipo
Amazon CloudWatch	com.amazonaws. <i>region-id</i> .registri	Interfaccia
CloudFormation	cn.com.amazonaws. <i>region-id</i> . formazione di nuvole	Interfaccia
Amazon EC2	cn.com.amazonaws. <i>region-id</i> .ec2	Interfaccia
Simple Storage Service (Amazon S3)	com.amazonaws.it. <i>region-id</i> .s3	Gateway
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	Gateway
Gestione dei segreti AWS**	com.amazonaws. <i>region-id</i> . gestore dei segreti	Interfaccia
AWS Elastic Load Balancing ***	com.amazonaws. <i>region-id</i> . bilanciamento elastico del carico	Interfaccia
AWS Scalabilità automatica***	cn.com.amazonaws. <i>region-id</i> .scalabilità automatica	Interfaccia

** Questo endpoint è richiesto solo quando è abilitato, altrimenti [DirectoryService](#) è facoltativo.

*** Questi endpoint sono necessari solo quando [LoginNodes](#) sono abilitati, altrimenti sono opzionali.

Tutte le istanze nel VPC devono disporre di gruppi di sicurezza adeguati per comunicare con gli endpoint. È possibile farlo aggiungendo gruppi di sicurezza nelle configurazioni [HeadNode](#) e [AdditionalSecurityGroupsAdditionalSecurityGroups](#) nelle configurazioni. [SlurmQueues](#)
Ad esempio, se gli endpoint VPC vengono creati senza specificare esplicitamente un gruppo di

sicurezza, il gruppo di sicurezza predefinito viene associato agli endpoint. Aggiungendo il gruppo di sicurezza predefinito `AdditionalSecurityGroups`, si abilita la comunicazione tra il cluster e gli endpoint.

Note

Quando utilizzi le policy IAM per limitare l'accesso agli endpoint VPC, devi aggiungere quanto segue all'endpoint VPC Amazon S3:

```
PolicyDocument:
  Version: 2012-10-17
  Statement:
    - Effect: Allow
      Principal: "*"
      Action:
        - "s3:PutObject"
      Resource:
        - !Sub "arn:${AWS::Partition}:s3::cloudformation-waitcondition-${AWS::Region}/*"
```

Disattiva Route 53 e usa i nomi host di Amazon EC2

Quando si crea un Slurm cluster, AWS ParallelCluster crea una zona ospitata privata sulla Route 53 che viene utilizzata per risolvere i nomi host dei nodi di calcolo personalizzati, ad esempio. `{queue_name}-{st|dy}-{compute_resource}-{N}`. Poiché Route 53 non supporta gli endpoint VPC, questa funzionalità deve essere disabilitata. Inoltre, AWS ParallelCluster deve essere configurato per utilizzare i nomi host predefiniti di Amazon EC2, ad esempio. `ip-1-2-3-4`. Applica le seguenti impostazioni alla configurazione del cluster:

```
...
Scheduling:
  ...
  SlurmSettings:
    Dns:
      DisableManagedDns: true
      UseEc2Hostnames: true
```

 Warning

Per i cluster creati con [SlurmSettings/Dns/DisableManagedDns](#) [UseEc2Hostnames](#) impostati su `true`, Slurm `nodeName` non viene risolto dal DNS. Usa invece. `Slurm NodeHostName`

 Note

Questa nota non è rilevante a partire dalla AWS ParallelCluster versione 3.3.0.
Per le versioni AWS ParallelCluster supportate precedenti alla 3.3.0:
Quando `UseEc2Hostnames` è impostato su `true`, il file Slurm di configurazione viene impostato con gli epilog script AWS ParallelCluster prolog and:

- `prolog` viene eseguito per aggiungere informazioni `/etc/hosts` sui nodi ai nodi di calcolo quando ogni lavoro viene allocato.
- `epilog` viene eseguito per pulire i contenuti scritti da `prolog`

Per aggiungere epilog script prolog o personalizzati, aggiungili rispettivamente alle `/opt/slurm/etc/pcluster/epilog.d/` cartelle `/opt/slurm/etc/pcluster/prolog.d/` o.

Configurazione del cluster

Scopri come configurare il cluster per l'esecuzione in una sottorete senza connessione a Internet.

La configurazione per questa architettura richiede le seguenti impostazioni:

```
# Note that all values are only provided as examples
...
HeadNode:
  ...
  Networking:
    SubnetId: subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
endpoints
    AdditionalSecurityGroups:
      - sg-abcdef01234567890 # optional, the security group that enables the
communication between the cluster and the VPC endpoints
```

```

LoginNodes: # optional, if enabled, requires creation and configuration of VPC
endpoints for AWS Elastic Load Balancing (ELB) and Auto Scaling services
Pools:
  - ...
  Networking:
    SubnetIds:
      - subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
endpoints attached
    AdditionalSecurityGroups:
      - sg-1abcdef01234567890 # optional, the security group that enables the
communication between the cluster and the VPC endpoints
Scheduling:
  Scheduler: Slurm # Cluster in a subnet without internet access is supported only when
the scheduler is Slurm.
  SlurmSettings:
    Dns:
      DisableManagedDns: true
      UseEc2Hostnames: true
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
        - subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
endpoints attached
      AdditionalSecurityGroups:
        - sg-1abcdef01234567890 # optional, the security group that enables the
communication between the cluster and the VPC endpoints

```

- [SubnetId\(s\)](#): La sottorete senza accesso a Internet.

Per abilitare la comunicazione tra AWS ParallelCluster e AWS Servizi, il VPC della sottorete deve avere gli endpoint VPC collegati. Prima di creare il cluster, verifica che l'[assegnazione automatica IPv4 dell'indirizzo pubblico sia disabilitata](#) nella sottorete per garantire che i `pcluster` comandi abbiano accesso al cluster.

- [AdditionalSecurityGroups](#): il gruppo di sicurezza che abilita la comunicazione tra il cluster e gli endpoint VPC.

Facoltativo:

- Se gli endpoint VPC vengono creati senza specificare esplicitamente un gruppo di sicurezza, viene associato il gruppo di sicurezza predefinito del VPC. Pertanto, fornisci il gruppo di sicurezza predefinito a. `AdditionalSecurityGroups`

- Se durante la creazione del cluster vengono utilizzati gruppi di sicurezza personalizzati, and/or `AdditionalSecurityGroups` gli endpoint VPC non sono necessari purché i gruppi di sicurezza personalizzati consentano la comunicazione tra il cluster e gli endpoint VPC.
- [Scheduler](#): Lo scheduler del cluster.

`slurm` è l'unico valore valido. Solo lo Slurm scheduler supporta un cluster in una sottorete senza accesso a Internet.

- [SlurmSettings](#): Le impostazioni. Slurm

Consulta la sezione precedente Disattiva Route53 e usa i nomi host di Amazon EC2.

Limitazioni

- Connessione al nodo principale tramite SSH o Amazon DCV: quando ti connetti a un cluster, assicurati che il client della connessione possa raggiungere il nodo principale del cluster tramite il suo indirizzo IP privato. Se il client non si trova nello stesso VPC del nodo principale, utilizza un'istanza proxy in una sottorete pubblica del VPC. Questo requisito si applica sia alle connessioni SSH che a quelle DCV. L'IP pubblico di un nodo principale non è accessibile se la sottorete non dispone di accesso a Internet. I `dcv-connect` comandi `pcluster ssh` and utilizzano l'IP pubblico, se esistente, o l'IP privato. Prima di creare il cluster, verifica che l'[assegnazione automatica IPv4 dell'indirizzo pubblico sia disabilitata](#) nella sottorete per garantire che i `pcluster` comandi abbiano accesso al cluster.

L'esempio seguente mostra come è possibile connettersi a una sessione DCV in esecuzione nel nodo principale del cluster. Ti connetti tramite un'istanza proxy Amazon EC2. L'istanza funziona come server Amazon DCV per il tuo PC e come client per il nodo principale nella sottorete privata.

Connect tramite DCV tramite un'istanza proxy in una sottorete pubblica:

1. Crea un'istanza Amazon EC2 in una sottorete pubblica, che si trova nello stesso VPC della sottorete del cluster.
2. Assicurati che il client e il server Amazon DCV siano installati sulla tua istanza Amazon EC2.
3. Allega una AWS ParallelCluster User Policy all'istanza proxy di Amazon EC2. Per ulteriori informazioni, consulta [AWS ParallelCluster esempi di politiche pcluster utente](#).
4. Installa AWS ParallelCluster sull'istanza proxy Amazon EC2.
5. Connect tramite DCV all'istanza proxy Amazon EC2.

6. Usa il `pcluster dcv-connect` comando sull'istanza proxy per connetterti al cluster all'interno della sottorete senza accesso a Internet.
- Interazione con altri AWS servizi: solo i servizi strettamente richiesti da AWS ParallelCluster sono elencati sopra. Se il cluster deve interagire con altri servizi, crea gli endpoint VPC corrispondenti.

Nodi di accesso forniti da AWS ParallelCluster

A partire dalla versione 3.7.0, gli amministratori del AWS ParallelCluster cluster possono fornire nodi di accesso che possono essere utilizzati per fornire agli utenti l'accesso all'esecuzione dei lavori anziché accedere direttamente al nodo principale del cluster. Gli utenti del cluster con le autorizzazioni appropriate possono utilizzare Active Directory o le proprie credenziali ssh per accedere, inviare e gestire i propri lavori. Di conseguenza, è possibile migliorare la gestione del cluster e ridurre al minimo le possibilità di esaurire le risorse del nodo principale necessarie Slurm per gestire il cluster. Gli utenti registrati avranno inoltre accesso a tutto lo storage condiviso del cluster montato sui nodi di accesso. Se è necessario interrompere un nodo di accesso, gli utenti che hanno effettuato l'accesso verranno avvisati in anticipo tramite la sessione di shell attiva che stanno utilizzando.

I nodi di accesso sono specificati come pool in cui un pool definisce un gruppo di nodi di accesso con la stessa configurazione di risorse. Tutti i nodi di accesso in un pool sono configurati per far parte di un [sistema di bilanciamento del carico di rete](#) che consente di distribuire le sessioni tra i nodi di accesso in modo ininterrotto. La presente implementazione consente agli utenti di specificare più pool di nodi di accesso.

Sicurezza per i nodi di accesso

[I nodi di accesso ereditano AllowedIPs le impostazioni AllowedIps dal nodo principale, a meno che non AllowedIps sia specificato per il pool di nodi di accesso.](#) In questo modo, gli amministratori del cluster possono limitare il livello di sicurezza del cluster specificando il CIDR di origine o un elenco di prefissi da cui sono consentite le connessioni SSH sul nodo principale o su un pool di nodi di accesso.

Nella presente implementazione, l'accesso al nodo principale non viene automaticamente limitato quando si abilitano i nodi di accesso. Se necessario, un amministratore del cluster può limitare questo accesso aggiornando la configurazione ssh dei nodi principali utilizzando i comandi Linux standard. Ciò può essere ottenuto anche specificando gruppi di sicurezza personalizzati sul nodo

principale utilizzando l'`AdditionalSecurityGroups` impostazione nella sezione `head node` del file `ParallelCluster` YAML per negare le connessioni da parte di utenti non autorizzati.

Rete per i nodi di accesso

Ai nodi di accesso viene assegnato un unico indirizzo di connessione al sistema di bilanciamento del carico di rete configurato per il pool di nodi di accesso. Le impostazioni di connettività dell'indirizzo si basano sul tipo di sottorete specificato nella configurazione del pool dei nodi di accesso.

- Se la sottorete è privata, l'indirizzo sarà privato e, per concedere l'accesso ai nodi di accesso, l'amministratore del cluster deve fornire un bastion host.
- Se la sottorete è pubblica, l'indirizzo sarà pubblico

Tutte le richieste di connessione sono gestite dal Network Load Balancer utilizzando il routing round-robin.

Archiviazione per i nodi di accesso

Tutto lo storage condiviso configurato sul cluster utilizzando lo storage gestito ParallelCluster incluso verrà montato su tutti i nodi di accesso.

Recupera le informazioni sui nodi di accesso

Per recuperare l'indirizzo della singola connessione fornita per accedere ai nodi di accesso, l'amministratore del cluster può eseguire il comando [describe-cluster](#). Il comando fornirà anche ulteriori informazioni sullo stato dei nodi di accesso.

I nodi di accesso sono un nuovo tipo di nodo supportato da ParallelCluster che può essere specificato con il [describe-cluster-instances](#) comando quando si interroga lo stato di un tipo di nodo specifico.

La disponibilità di un singolo indirizzo di connessione al pool di nodi di accesso non impedisce l'accesso diretto a un nodo di accesso specifico. Tuttavia, non è consigliabile utilizzare la connessione diretta per evitare avvisi dal client ssh. Il client ssh memorizza gli identificatori host localmente per ogni indirizzo di destinazione. Poiché l'identificatore host è specifico per pool, l'uso di indirizzi di connessione diversi IPs e/o singoli può avere lo stesso identificatore host associato a diversi indirizzi di destinazione: ciò può causare un avviso da parte del client ssh poiché allo stesso identificatore host sono associati più destinazioni.

Proprietà Imds per i nodi di accesso

L'accesso all'IMDS del nodo di accesso (e alle credenziali del profilo di istanza) è limitato all'utente root, all'utente amministrativo del cluster (`pc-cluster-admin` per impostazione predefinita) e all'utente predefinito specifico del sistema operativo (`ubuntu` `ec2-user` Amazon Linux 2 e RedHat Ubuntu 18.04).

Per limitare l'accesso IMDS, AWS ParallelCluster gestisce una catena di `iptables`

Note

Qualsiasi personalizzazione `iptables` o `ip6tables` regola può interferire con il meccanismo utilizzato per limitare l'accesso IMDS sul nodo di accesso. Vedi anche. [Imds property setting](#)

Ciclo di vita dei nodi di accesso

Attualmente non esiste un comando dedicato per arrestare e avviare i nodi di accesso in un pool. Per interrompere i nodi di accesso in un pool, l'amministratore del cluster deve aggiornare la configurazione del cluster specificando zero sul numero di nodi di accesso (`Count: 0`) e quindi eseguire un [pcluster.update-cluster-v3](#) comando.

Note

Gli utenti che hanno effettuato l'accesso vengono informati della chiusura dell'istanza specifica e del relativo periodo di grazia. [Durante il periodo di grazia non saranno consentite nuove connessioni ad eccezione di quelle dell'utente predefinito del cluster.](#) Il messaggio mostrato è personalizzabile dall'amministratore del cluster dal nodo principale o da un nodo di accesso che modifica il file. `/opt/parallelcluster/shared_login_nodes/loginmgt_config.json` Questo messaggio di terminazione non è visibile quando ci si connette tramite [AWS Systems Manager Session Manager](#) Session Manager.

Per avviare il pool di nodi di accesso, l'amministratore del cluster deve ripristinare il `Count` valore precedente nella configurazione del cluster e quindi eseguire un [update-cluster](#) comando.

Autorizzazioni necessarie per eseguire il pool di nodi di accesso

Per gestire il pool di nodi di accesso, l'amministratore del cluster deve disporre delle seguenti autorizzazioni aggiuntive:

```
- Action:
  - iam:CreateServiceLinkedRole
  - autoscaling:DeleteAutoScalingGroup
  - autoscaling:DeleteLifecycleHook
  - autoscaling:Describe*
  - autoscaling:PutLifecycleHook
  - autoscaling:UpdateAutoScalingGroup
  - elasticloadbalancing:CreateListener
  - elasticloadbalancing:CreateTargetGroup
  - elasticloadbalancing>DeleteListener
  - elasticloadbalancing>DeleteLoadBalancer
  - elasticloadbalancing>DeleteTargetGroup
  - elasticloadbalancing:Describe*
  - elasticloadbalancing:ModifyLoadBalancerAttributes
Resource: '*'
Condition:
  ForAllValues:StringEquals:
    aws:TagKeys: [ "parallelcluster:cluster-name" ]
- Action:
  - autoscaling:CreateAutoScalingGroup
  - autoscaling>DeleteTags
  - autoscaling>CreateOrUpdateTags
  - elasticloadbalancing:AddTags
  - elasticloadbalancing>CreateLoadBalancer
  - elasticloadbalancing:RemoveTags
  - elasticloadbalancing:ModifyTargetGroup
Resource: '*'
Effect: Allow
```

Azioni bootstrap personalizzate

Se si definiscono le impostazioni di [OnNodeStart](#) configurazione [HeadNodeCustomActions](#)//, AWS ParallelCluster esegue codice arbitrario immediatamente dopo l'avvio del nodo. Se si definiscono le impostazioni di [OnNodeConfigured](#) configurazione [HeadNodeCustomActions](#)//,

AWS ParallelCluster esegue il codice dopo che la configurazione del nodo è stata completata correttamente.

A partire dalla AWS ParallelCluster versione 3.4.0, il codice può essere eseguito dopo l'aggiornamento del nodo principale, se si definiscono le impostazioni di [OnNodeUpdated](#) configurazione [HeadNodeCustomActions](#)//.

Nella maggior parte dei casi, questo codice è memorizzato in Amazon Simple Storage Service (Amazon S3) e vi si accede tramite una connessione HTTPS. Il codice viene eseguito root e può essere in qualsiasi linguaggio di script supportato dal sistema operativo del cluster. Spesso il codice è in Bash o Python.

Note

A partire dalla AWS ParallelCluster versione 3.7.0, l'impostazione [ImdsImdsSupport](#)cluster/ predefinita è. v2.0

Quando crei un nuovo cluster per l'aggiornamento alla versione 3.7.0 e versioni successive, aggiorna gli script di azione bootstrap personalizzati per renderli compatibili con IMDSv2 o imposta [Imds/ImdsSupport](#) nel file di configurazione del cluster. v1.0

Warning

[È responsabilità dell'utente configurare gli script e gli argomenti personalizzati come descritto nel modello di responsabilità condivisa.](#) Verifica che gli script e gli argomenti di bootstrap personalizzati provengano da fonti attendibili per avere accesso completo ai nodi del cluster.

Warning

AWS ParallelCluster non supporta l'uso di variabili interne fornite tramite il `/etc/parallelcluster/cfnconfig` file. Questo file potrebbe essere rimosso come parte di future release.

OnNodeStart le azioni vengono richiamate prima dell'avvio di qualsiasi azione di bootstrap di distribuzione del nodo, come la configurazione di NAT, Amazon Elastic Block Store (Amazon EBS)

o lo scheduler. `OnNodeStart`Le azioni di bootstrap possono includere la modifica dello storage, l'aggiunta di utenti aggiuntivi e l'aggiunta di pacchetti.

Note

Se [DirectoryService](#) configuri uno [OnNodeStart](#) script [HeadNode/CustomActions](#)/per il tuo cluster, AWS ParallelCluster configura DirectoryService e riavvia lo script prima che venga eseguito lo script. `OnNodeStart`

`OnNodeConfigured`Le azioni vengono richiamate dopo il completamento dei processi di bootstrap del nodo. `OnNodeConfigured`Le azioni servono alle ultime azioni che si verificano prima che un'istanza venga considerata completamente configurata e completa. Alcune `OnNodeConfigured` azioni includono la modifica delle impostazioni dello scheduler, la modifica dell'archiviazione e la modifica dei pacchetti. È possibile passare argomenti agli script specificandoli durante la configurazione.

`OnNodeUpdated`Le azioni vengono richiamate dopo il completamento dell'aggiornamento del nodo principale e l'allineamento dello scheduler e dello storage condiviso alle ultime modifiche alla configurazione del cluster.

Quando `OnNodeStart` le nostre azioni `OnNodeConfigured` personalizzate hanno esito positivo, il successo viene indicato con il codice di uscita zero (0). Qualsiasi altro codice di uscita indica che il bootstrap dell'istanza non è riuscito.

Quando le azioni `OnNodeUpdated` personalizzate hanno esito positivo, il successo viene segnalato con il codice di uscita zero (0). Qualsiasi altro codice di uscita indica che l'aggiornamento non è riuscito.

Note

Se si configura [OnNodeUpdated](#), è necessario ripristinare manualmente le `OnNodeUpdated` azioni allo stato precedente in caso di errori di aggiornamento.

Se un'azione `OnNodeUpdated` personalizzata fallisce, l'aggiornamento torna allo stato precedente. Tuttavia, l'`OnNodeUpdated`azione viene eseguita solo al momento dell'aggiornamento e non al momento del rollback dello stack.

È possibile specificare diversi script per il nodo principale e per ogni coda, nelle sezioni di configurazione [HeadNode/CustomActions](#) e i [Scheduling/SlurmQueues. CustomActions](#) [OnNodeUpdated](#) può essere configurato solo nella HeadNode sezione.

Note

Prima della AWS ParallelCluster versione 3.0, non era possibile specificare script diversi per i nodi head e compute. Fare riferimento a [Passaggio da AWS ParallelCluster 2.x a 3.x](#).

Argomenti

- [Impostazioni di configurazione per definire azioni e argomenti](#)
- [Arguments \(Argomenti\)](#)
- [Cluster di esempio con azioni bootstrap personalizzate](#)
- [Esempio di aggiornamento di uno script di bootstrap personalizzato per IMDSv2](#)
- [Esempio di come aggiornare una configurazione per IMDSv1](#)

Impostazioni di configurazione per definire azioni e argomenti

Le seguenti impostazioni di configurazione vengono utilizzate per definire [HeadNodeOnNodeConfigured](#) azioni [OnNodeStart](#) [OnNodeConfigured](#) argomenti [Scheduling/CustomActions/OnNodeStart](#) & & e//. [CustomActionsOnNodeUpdated](#)

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URL. This is run before any of the bootstrap scripts are run
      Script: s3://amzn-s3-demo-bucket/on-node-start.sh
      Args:
        - arg1
    OnNodeConfigured:
      # Script URL. This is run after all the bootstrap scripts are run
      Script: s3://amzn-s3-demo-bucket/on-node-configured.sh
      Args:
        - arg1
    OnNodeUpdated:
      # Script URL. This is run after the head node update is completed.
```

```

    Script: s3://amzn-s3-demo-bucket/on-node-updated.sh
    Args:
      - arg1
  # Bucket permissions
  Iam:
    S3Access:
      - BucketName: bucket_name
      EnableWriteAccess: false
  Scheduling:
    Scheduler: slurm
    [...]
  SlurmQueues:
    - Name: queue1
    [...]
  CustomActions:
    OnNodeStart:
      Script: s3://amzn-s3-demo-bucket/on-node-start.sh
      Args:
        - arg1
    OnNodeConfigured:
      Script: s3://amzn-s3-demo-bucket/on-node-configured.sh
      Args:
        - arg1
  Iam:
    S3Access:
      - BucketName: bucket_name
      EnableWriteAccess: false

```

Utilizzando l'Sequence impostazione (aggiunta nella AWS ParallelCluster versione 3.6.0):

```

HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, before any of the bootstrap scripts are run.
      Sequence:
        - Script: s3://amzn-s3-demo-bucket/on-node-start1.sh
          Args:
            - arg1
        - Script: s3://amzn-s3-demo-bucket/on-node-start2.sh
          Args:
            - arg1

```



```

    [...]
    OnNodeConfigured:
        # Script URLs. The scripts are run in the same order as listed in the
        configuration, after all the bootstrap scripts are run.
        Sequence:
            - Script: s3://amzn-s3-demo-bucket/on-node-configured1.sh
              Args:
                - arg1
            - Script: s3://amzn-s3-demo-bucket/on-node-configured2.sh
              Args:
                - arg1
        [...]
    OnNodeUpdated:
        # Script URLs. The scripts are run in the same order as listed in the
        configuration, after the head node update is completed.
        Sequence:
            - Script: s3://amzn-s3-demo-bucket/on-node-updated1.sh
              Args:
                - arg1
            - Script: s3://amzn-s3-demo-bucket/on-node-updated2.sh
              Args:
                - arg1
        [...]
    # Bucket permissions
    Iam:
        S3Access:
            - BucketName: bucket_name
              EnableWriteAccess: false
    Scheduling:
        Scheduler: slurm
        [...]
    SlurmQueues:
        - Name: queue1
        [...]
        CustomActions:
            OnNodeStart:
                # Script URLs. The scripts are run in the same order as listed in the
                configuration, before any of the bootstrap scripts are run
                Sequence:
                    - Script: s3://amzn-s3-demo-bucket/on-node-start1.sh
                      Args:
                        - arg1
                    - Script: s3://amzn-s3-demo-bucket/on-node-start2.sh
                      Args:

```

```

        - arg1
    [...]
    OnNodeConfigured:
        # Script URLs. The scripts are run in the same order as listed in the
        configuration, after all the bootstrap scripts are run
        Sequence:
            - Script: s3://amzn-s3-demo-bucket/on-node-configured1.sh
              Args:
                - arg1
            - Script: s3://amzn-s3-demo-bucket/on-node-configured2.sh
              Args:
                - arg1
    [...]
    Iam:
        S3Access:
            - BucketName: bucket_name
              EnableWriteAccess: false

```

L'azione `Sequence` viene aggiunta a partire dalla AWS ParallelCluster versione 3.6.0. Quando si specifica `Sequence`, è possibile elencare più script per un'azione personalizzata. AWS ParallelCluster continua a supportare la configurazione di un'azione personalizzata con un singolo script, senza includere `Sequence`.

AWS ParallelCluster non supporta l'inclusione sia di un singolo script che della stessa azione personalizzata. Ad esempio, AWS ParallelCluster fallisce se si specifica la seguente configurazione.

```

[...]
CustomActions:
    OnNodeStart:
        # Script URL. This is run before any of the bootstrap scripts are run
        Script: s3://amzn-s3-demo-bucket/on-node-start.sh
        Args:
            - arg1
        # Script URLs. The scripts are run in the same order as listed in the
        configuration, before any of the bootstrap scripts are run.
        Sequence:
            - Script: s3://amzn-s3-demo-bucket/on-node-start1.sh
              Args:
                - arg1
            - Script: s3://amzn-s3-demo-bucket/on-node-start2.sh
              Args:
                - arg1

```

[...]

Arguments (Argomenti)

In AWS ParallelCluster 2.x gli \$1 argomenti erano riservati, per memorizzare l'URL dello script personalizzato. Se vuoi riutilizzare gli script di bootstrap personalizzati creati per AWS ParallelCluster 2.x con AWS ParallelCluster 3.x, devi adattarli considerando lo spostamento degli argomenti. Fare riferimento a [Passaggio da AWS ParallelCluster 2.x a 3.x](#).

Cluster di esempio con azioni bootstrap personalizzate

I passaggi seguenti creano un semplice script da eseguire dopo la configurazione del nodo, che installa i wget pacchetti R, curl e nei nodi del cluster.

1. Creare uno script.

```
#!/bin/bash
echo "The script has $# arguments"
for arg in "$@"
do
    echo "arg: ${arg}"
done
yum -y install "${@:1}"
```

2. Carica lo script con le autorizzazioni corrette su Amazon S3. Se le autorizzazioni di lettura pubbliche non sono appropriate per te, usa le sezioni [HeadNode/IamScheduling/S3AccessSlurmQueues](#) /configuration. Per ulteriori informazioni, consulta [Lavorare con Amazon S3](#).

```
$ aws s3 cp --acl public-read /path/to/myscript.sh s3://amzn-s3-demo-bucket/myscript.sh
```

Important

Se lo script è stato modificato in Windows, le terminazioni di riga devono essere modificate da CRLF a LF prima di caricare lo script su Amazon S3.

3. Aggiorna la AWS ParallelCluster configurazione per includere la nuova azione. OnNodeConfigured

```
CustomActions:
  OnNodeConfigured:
    Script: https://<amzn-s3-demo-bucket>.s3.<region>.amazonaws.com/myscript.sh
    Args:
      - "R"
      - "curl"
      - "wget"
```

Se il bucket non dispone dell'autorizzazione di lettura pubblica, utilizzalo s3 come protocollo URL.

```
CustomActions:
  OnNodeConfigured:
    Script: s3://<amzn-s3-demo-bucket>/myscript.sh
    Args:
      - "R"
      - "curl"
      - "wget"
```

4. Avviare il cluster

```
$ pcluster create-cluster --cluster-name mycluster \
  --region <region> --cluster-configuration config-file.yaml
```

5. Verificare l'output.

- Se hai aggiunto azioni personalizzate alla HeadNode configurazione, accedi al nodo head e controlla il `cfn-init.log` file in cui si trova `/var/log/cfn-init.log` eseguendo il seguente comando:

```
$ less /var/log/cfn-init.log
2021-09-03 10:43:54,588 [DEBUG] Command run
postinstall output: The script has 3 arguments
arg: R
arg: curl
arg: wget
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

- Se hai aggiunto azioni personalizzate all'SlurmQueuesimpostazione, controlla l'indirizzo `cloud-init.log` situato `/var/log/cloud-init.log` in un nodo di calcolo. CloudWatch Utilizzatelo per visualizzare questi registri.

Puoi visualizzare entrambi questi log nella CloudWatch console Amazon. Per ulteriori informazioni, consulta [Integrazione con Amazon CloudWatch Logs](#).

Esempio di aggiornamento di uno script di bootstrap personalizzato per IMDSv2

Nell'esempio seguente, aggiorniamo uno script di azione bootstrap personalizzato utilizzato con IMDSv1 per l'utilizzo con IMDSv2. Lo script IMDSv1 recupera i metadati dell'ID AMI dell'istanza Amazon EC2.

```
#!/bin/bash
AMI_ID=$(curl http://169.254.169.254/latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

Di seguito viene illustrato lo script di azione bootstrap personalizzato modificato per renderlo compatibile con. IMDSv2

```
#!/bin/bash
AMI_ID=$(TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

Per ulteriori informazioni, consulta [Recupera i metadati dell'istanza](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Esempio di come aggiornare una configurazione per IMDSv1

Di seguito è riportato un esempio di configurazione del cluster che supporta l' IMDSv1 utilizzo delle AWS ParallelCluster versioni 3.7.0 e precedenti.

```
Region: us-east-1
Imds:
```

```
ImdsSupport: v1.0
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: key-name
  CustomActions:
    OnNodeConfigured:
      Script: Script-path
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CustomActions:
        OnNodeConfigured:
          Script: Script-path
      ComputeResources:
        - Name: t2micro
          Instances:
            - InstanceType: t2.micro
          MinCount: 11
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
```

Lavorare con Amazon S3

Puoi configurare AWS ParallelCluster l'accesso ad Amazon S3 tramite i [S3Access](#) parametri [HeadNode/Iam/S3Access](#) [Scheduling/SlurmQueues/- Name/Iam](#) nella AWS ParallelCluster configurazione.

Esempi

L'esempio seguente configura l'accesso in sola lettura a tutti gli oggetti in *firstbucket/read_only/* e l'accesso in lettura/scrittura a tutti gli oggetti in. *secondbucket/read_and_write/*

```
...
HeadNode:
  ...
```

```
Iam:
  S3Access:
    - BucketName: firstbucket
      KeyName: read_only/*
      EnableWriteAccess: false
    - BucketName: secondbucket
      KeyName: read_and_write/*
      EnableWriteAccess: true
  ...
```

L'esempio successivo configura l'accesso in sola lettura a tutti gli oggetti nella cartella *read_only/* in qualsiasi bucket (*) dell'account.

```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: *
        KeyName: read_only/*
        EnableWriteAccess: false
  ...
```

L'ultimo esempio configura l'accesso in sola lettura a tutti i bucket e gli oggetti dell'account.

```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: *
  ...
```

Utilizzo di Istanze spot

AWS ParallelCluster utilizza le istanze Spot se hai impostato [SlurmQueues/CapacityType](#) o [AwsBatchQueues/CapacityType](#) a SPOT nel file di configurazione del cluster. Le istanze Spot sono più convenienti rispetto alle istanze on demand, ma potrebbero subire interruzioni. Potrebbe essere utile sfruttare gli avvisi di interruzione dell'istanza Spot, che forniscono un avviso di due minuti prima che Amazon EC2 debba interrompere o terminare l'istanza Spot. Per ulteriori informazioni,

consulta le [interruzioni delle istanze Spot](#) nella Guida per l'utente di Amazon EC2. Per scoprire come [AwsBatchQueues](#) funziona con le istanze Spot, consulta [Compute Resources](#) nella Guida per l'utente. AWS Batch

Lo scheduler AWS ParallelCluster configurato assegna i lavori alle risorse di calcolo in code con istanze Spot nello stesso modo in cui assegna i lavori alle risorse di calcolo in code con istanze on demand.

Quando utilizzi le istanze Spot, nel tuo account deve esistere un ruolo collegato al servizio Spot. AWSService RoleFor EC2 Per creare questo ruolo nel tuo account utilizzando AWS CLI, esegui il seguente comando:

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Per ulteriori informazioni, consulta il [ruolo collegato ai servizi per le richieste di istanze Spot nella Guida](#) per l'utente di Amazon EC2.

Le sezioni seguenti descrivono tre scenari in cui le istanze Spot possono essere interrotte durante l'utilizzo. [SlurmQueues](#)

Scenario 1: viene interrotta un'istanza Spot senza attività in esecuzione

Quando si verifica questa interruzione, AWS ParallelCluster tenta di sostituire l'istanza se la coda dello scheduler ha lavori in sospeso che richiedono istanze aggiuntive o se il numero di istanze attive è inferiore a//. [SlurmQueuesComputeResourcesMinCount](#) Se non è AWS ParallelCluster possibile fornire nuove istanze, viene ripetuta periodicamente una richiesta di nuove istanze.

Scenario 2: l'istanza spot che esegue attività a nodo singolo viene interrotta

Il processo ha esito negativo con un codice di stato pari a `NODE_FAIL`, e il processo viene richiesto (a meno che non `--no-requeue` sia specificato al momento dell'invio del lavoro). Se il nodo è statico, viene sostituito. Se il nodo è un nodo dinamico, viene terminato e reimpostato. Per ulteriori informazioni sull'`sbatch` inclusione del `--no-requeue` parametro, [sbatch](#) consultate la Slurmdocumentazione.

Scenario 3: l'istanza spot che esegue attività a più nodi viene interrotta

Il processo ha esito negativo con un codice di stato pari a `NODE_FAIL`, e il processo viene richiesto (a meno che non sia `--no-requeue` stato specificato al momento dell'invio del lavoro). Se il nodo

è statico, viene sostituito. Se il nodo è un nodo dinamico, viene terminato e reimpostato. Gli altri nodi che eseguivano i processi terminati potrebbero essere assegnati ad altri lavori in sospeso o ridimensionati una volta trascorso il tempo/configurato [SlurmSettings](#). [ScaledownIdleTime](#)

Per ulteriori informazioni sulle istanze Spot, consulta le istanze [Spot nella Guida](#) per l'utente di Amazon EC2.

Scheduler supportati da AWS ParallelCluster

AWS ParallelCluster supporta Slurm e AWS Batch pianificatori, che vengono impostati utilizzando l'impostazione. [Scheduler](#) I seguenti argomenti descriveranno ogni scheduler e come utilizzarli.

Argomenti

- [Slurm Workload Manager \(slurm\)](#)
- [Utilizzo dello scheduler AWS Batch \(awsbatch\) con AWS ParallelCluster](#)

Slurm Workload Manager (**slurm**)

Dimensione e aggiornamento della capacità del cluster

La capacità del cluster è definita dal numero di nodi di calcolo scalabili dal cluster. I nodi di calcolo sono supportati da istanze Amazon EC2 definite all'interno delle risorse di calcolo nella (Scheduling/SlurmQueues/[ComputeResources](#)) configurazione e sono organizzati in AWS ParallelCluster (Scheduling/[SlurmQueues](#)) code che mappano 1:1 alle partizioni. Slurm

[All'interno di una risorsa di elaborazione è possibile configurare il numero minimo di nodi di calcolo \(istanze\) che devono essere sempre mantenuti in esecuzione nel cluster \(\) e il numero massimo di istanze fino a cui la risorsa di calcolo può scalare \(3MinCount\). MaxCount](#)

Al momento della creazione del cluster o dopo un aggiornamento del cluster, AWS ParallelCluster avvia tutte le istanze Amazon EC2 configurate per ogni risorsa di calcolo Scheduling/SlurmQueues/ [ComputeResources](#) () definita MinCount nel cluster. Le istanze avviate per coprire il numero minimo di nodi per le risorse di calcolo nel cluster sono chiamate nodi statici. Una volta avviati, i nodi statici sono pensati per essere persistenti nel cluster e non vengono terminati dal sistema, a meno che non si verifichi un evento o una condizione particolare. Tali eventi includono, ad esempio, il fallimento dei controlli di Slurm integrità di Amazon EC2 e la modifica dello stato del Slurm nodo in DRAIN o DOWN.

Le istanze Amazon EC2, nell'intervallo da 1 **1** a '**MaxCount** - **MinCount**' (**MaxCount** meno **MinCount**), lanciate su richiesta per far fronte all'aumento del carico del cluster, sono denominate nodi dinamici. La loro natura è effimera, vengono avviate per eseguire lavori in sospeso e vengono terminate quando rimangono inattive per un periodo di tempo definito `Scheduling/SlurmSettings/ScaledownIdleTime` nella configurazione del cluster (impostazione predefinita: 10 minuti).

I nodi statici e i nodi dinamici sono conformi al seguente schema di denominazione:

- Nodi statici dove `<Queue/Name>-st-<ComputeResource/Name>-<num> <num> = 1..ComputeResource/MinCount`
- Nodi dinamici `<Queue/Name>-dy-<ComputeResource/Name>-<num>` dove `<num> = 1..(ComputeResource/MaxCount - ComputeResource/MinCount)`

Ad esempio, data la seguente AWS ParallelCluster configurazione:

```
Scheduling:
  Scheduler: Slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

I seguenti nodi verranno definiti in Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up      infinite   100    idle queue1-st-c5xlarge-[1-100]
```

Quando una risorsa di elaborazione lo ha `MinCount == MaxCount`, tutti i nodi di calcolo corrispondenti saranno statici e tutte le istanze verranno avviate al creation/update momento del cluster e mantenute attive e funzionanti. Esempio:

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
          MinCount: 100
          MaxCount: 100
```

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    100   idle queue1-st-c5xlarge-[1-100]
```

Aggiornamento della capacità del cluster

L'aggiornamento della capacità del cluster include l'aggiunta o la rimozione di code, risorse di elaborazione o la modifica MinCount/MaxCount di una risorsa di elaborazione. A partire dalla AWS ParallelCluster versione 3.9.0, la riduzione delle dimensioni di una coda richiede che il parco di elaborazione venga interrotto o [QueueUpdateStrategy](#) impostato su TERMINATE prima dell'aggiornamento del cluster. Non è necessario interrompere la flotta di elaborazione o impostare su TERMINATE quando: [QueueUpdateStrategy](#)

- Aggiungere nuove code a Scheduling/ [SlurmQueues](#)
- Aggiungere nuove risorse di calcolo a una coda Scheduling/ [SlurmQueues/ComputeResources](#)
- Aumento della potenza [MaxCount](#) di una risorsa di elaborazione
- Aumento MinCount di una risorsa di elaborazione e aumento MaxCount della stessa risorsa di elaborazione almeno della stessa quantità

Considerazioni e limitazioni

Questa sezione ha lo scopo di delineare eventuali fattori, vincoli o limitazioni importanti da tenere in considerazione durante il ridimensionamento della capacità del cluster.

- Quando si rimuove una coda da `Scheduling/SlurmQueues` tutti i nodi di elaborazione con nome `<Queue/Name>-*`, sia statici che dinamici, verrà rimossa dalla Slurm configurazione e le istanze Amazon EC2 corrispondenti verranno terminate.
- Quando si rimuove una risorsa di calcolo `Scheduling/SlurmQueues/ComputeResources` da una coda, tutti i nodi di calcolo con nome `<Queue/Name>-*-<ComputeResource/Name>-*`, statici e dinamici, verranno rimossi dalla Slurm configurazione e le istanze Amazon EC2 corrispondenti verranno terminate.

Quando si modifica il `MinCount` parametro di una risorsa di calcolo, possiamo distinguere due diversi scenari, se `MaxCount` viene mantenuto uguale a `MinCount` (solo capacità statica) e se `MaxCount` è maggiore `MinCount` di (capacità statica e dinamica mista).

La capacità cambia solo con nodi statici

- Se `MinCount == MaxCount`, aumentando `MinCount` (e `MaxCount`), il cluster verrà configurato estendendo il numero di nodi statici al nuovo valore di `MinCount` `<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` e il sistema continuerà a provare ad avviare istanze Amazon EC2 per soddisfare la nuova capacità statica richiesta.
- Se `MinCount == MaxCount`, diminuendo `MinCount` (e `MaxCount`) la quantità `N`, il cluster verrà configurato rimuovendo gli ultimi `N` nodi statici `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>` e il sistema interromperà le istanze Amazon EC2 corrispondenti.

- Stato iniziale `MinCount = MaxCount = 100`

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    100   idle queue1-st-c5xlarge-[1-100]
```

- Aggiornamento -30 su `MinCount` e `MaxCount`: `MinCount = MaxCount = 70`

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    70    idle queue1-st-c5xlarge-[1-70]
```

Modifiche di capacità con nodi misti

Se $\text{MinCount} < \text{MaxCount}$, aumentando MinCount di un valore N (supponendo che MaxCount venga mantenuto invariato), il cluster verrà configurato estendendo il numero di nodi statici al nuovo valore di MinCount ($\text{old\_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N>` e il sistema continuerà a provare ad avviare istanze Amazon EC2 per soddisfare la nuova capacità statica richiesta. Inoltre, per rispettare la MaxCount capacità della risorsa di elaborazione, la configurazione del cluster viene aggiornata rimuovendo gli ultimi N nodi dinamici `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old\_MinCount - N>...<MaxCount - old\_MinCount>]` e il sistema interromperà le istanze Amazon EC2 corrispondenti.

- Stato iniziale: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite    100   idle queue1-st-c5xlarge-[1-100]
```

- Aggiorna +30 a MinCount : $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*    up       infinite    130   idle queue1-st-c5xlarge-[1-130]
```

Se $\text{MinCount} < \text{MaxCount}$, aumentando MinCount e MaxCount della stessa quantità N , il cluster verrà configurato estendendo il numero di nodi statici al nuovo valore di MinCount (old_MinCount

+ N): <Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N> e il sistema continuerà a provare ad avviare istanze Amazon EC2 per soddisfare la nuova capacità statica richiesta. Inoltre, non verrà apportata alcuna modifica al numero di nodi dinamici per rispettare il nuovo

MaxCount value.

- Stato iniziale: MinCount = 100; MaxCount = 150

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Aggiorna +30 a MinCount : MinCount = 130 (MaxCount = 180)

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   130    idle queue1-st-c5xlarge-[1-130]
```

Se $\text{MinCount} < \text{MaxCount}$, diminuendo la MinCount quantità N (supponendo che MaxCount venga mantenuta invariata), il cluster verrà configurato rimuovendo gli ultimi N nodi statici dei nodi statici <Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>] e il sistema interromperà le istanze Amazon EC2 corrispondenti. Inoltre, per rispettare la MaxCount capacità della risorsa di elaborazione, la configurazione del cluster viene aggiornata estendendo il numero di nodi dinamici per colmare il divario. $\text{MaxCount} - \text{new\_MinCount}$: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>] In questo caso, trattandosi di nodi dinamici, non verranno lanciate nuove istanze Amazon EC2 a meno che lo scheduler non abbia lavori in sospeso sui nuovi nodi.

- Stato iniziale: MinCount = 100; MaxCount = 150

-

```
$ sinfo
```

```

PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up      infinite   100    idle queue1-st-c5xlarge-[1-100]

```

- Aggiornamento -30 su MinCount : MinCount = 70 (MaxCount = 120)

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    80    idle~ queue1-dy-c5xlarge-[1-80]
queue1*    up      infinite    70    idle queue1-st-c5xlarge-[1-70]

```

Se $\text{MinCount} < \text{MaxCount}$, in diminuzione MinCount e MaxCount della stessa quantità N, il cluster verrà configurato rimuovendo gli ultimi N nodi statici `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] e il sistema interromperà le istanze Amazon EC2 corrispondenti.

Inoltre, non verrà apportata alcuna modifica al numero di nodi dinamici per rispettare il nuovo valore. MaxCount

- Stato iniziale: MinCount = 100; MaxCount = 150

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up      infinite   100    idle queue1-st-c5xlarge-[1-100]

```

- Aggiornamento -30 su MinCount : MinCount = 70 (MaxCount = 120)

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    80    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up      infinite    70    idle queue1-st-c5xlarge-[1-70]

```

Se `MinCount < MaxCount`, diminuendo la `MaxCount` quantità `N` (supponendo `MinCount` che venga mantenuta invariata), il cluster verrà configurato rimuovendo gli ultimi `N` nodi dinamici `<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>]` e il sistema interromperà le istanze Amazon EC2 corrispondenti nel caso in cui fossero in esecuzione. Non è previsto alcun impatto sui nodi statici.

- Stato iniziale: `MinCount = 100`; `MaxCount = 150`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Aggiornamento -30 su `MaxCount` : `MinCount = 100` (`MaxCount = 120`)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

Impatti sui lavori

In tutti i casi in cui i nodi vengono rimossi e le istanze Amazon EC2 terminate, un job sbatch in esecuzione sui nodi rimossi verrà nuovamente messo in coda, a meno che non vi siano altri nodi che soddisfino i requisiti del processo. In quest'ultimo caso, il processo ha esito negativo con lo stato `NODE_FAIL` e scompare dalla coda e deve essere inviato nuovamente manualmente.

Se si prevede di eseguire un aggiornamento per il ridimensionamento del cluster, è possibile impedire l'esecuzione dei lavori nei nodi che verranno rimossi durante l'aggiornamento pianificato. Ciò è possibile impostando i nodi da rimuovere durante la manutenzione. Tieni presente che l'impostazione di un nodo in manutenzione non influirebbe sui lavori che alla fine sono già in esecuzione nel nodo.

Supponiamo che con l'aggiornamento pianificato per il ridimensionamento del cluster si voglia rimuovere il nodo `queueu-st-computeresource-[9-10]`. È possibile creare una Slurm prenotazione con il seguente comando

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queueu-st-
computeresource-[9-10]
```

Questo creerà una Slurm prenotazione denominata `maint_for_update` sui nodi `queueu-st-computeresource-[9-10]`. Dal momento in cui viene creata la prenotazione, non è possibile eseguire altri lavori nei nodi `queueu-st-computeresource-[9-10]`. Tieni presente che la prenotazione non impedirà l'assegnazione finale dei lavori sui nodi `queueu-st-computeresource-[9-10]`.

Dopo l'aggiornamento del ridimensionamento del cluster, se la Slurm prenotazione è stata impostata solo sui nodi che sono stati rimossi durante l'aggiornamento del ridimensionamento, la prenotazione di manutenzione verrà automaticamente eliminata. Se invece hai creato una Slurm prenotazione sui nodi che sono ancora presenti dopo l'aggiornamento del ridimensionamento del cluster, potremmo voler rimuovere la prenotazione di manutenzione sui nodi dopo l'esecuzione dell'aggiornamento di ridimensionamento, utilizzando il seguente comando

```
sudo -i scontrol delete ReservationName=maint_for_update
```

[Per ulteriori dettagli sulla Slurm prenotazione, consulta il documento ufficiale di SchedMD qui.](#)

Processo di aggiornamento del cluster in caso di modifiche alla capacità

In caso di modifica della configurazione dello scheduler, durante il processo di aggiornamento del cluster vengono eseguiti i seguenti passaggi:

- Interrompi AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- Genera una configurazione aggiornata Slurm delle partizioni dalla AWS ParallelCluster configurazione
- Riavvio `slurmctld` (eseguito tramite la ricetta del servizio Chef)
- Controlla `slurmctld` lo stato (`systemctl is-active --quiet slurmctld.service`)
- Ricarica la configurazione Slurm (`scontrol reconfigure`)
- Avvia `clustermgtd` (`supervisorctl start clustermgtd`)

Per informazioni su Slurm, consulta <https://slurm.schedmd.com>. Per i download, vedi <https://github.com/SchedMD/slurm/tags>. Per il codice sorgente, vedi slurm. <https://github.com/SchedMD/>

Versioni cluster e SLURM supportate

Nella tabella seguente sono elencate le Slurm versioni AWS ParallelCluster e supportate AWS .

AWS ParallelCluster versione (e)	Versione di Slurm supportata
3.13.0	24,05,07
3.12.0	23,11,10
3.11.0	23,11,10
3.9.2, 3.9.3, 3.10.0	23,11,7
3,9,0, 3,9,1	23,11,4
3.8.0	23,02,7
3.7.2	23,02,6
3.7.1	23,02,5
3.7.0	23,02,4
3.6.0, 3.6.1	23,02,2
3,5,0, 3,5,1	22,05,8
3.4.0, 3.4.1	22,05-7
3.3.0, 3.3.1	22,05,5
3.1.4, 3.1.5, 3.2.0, 3.2.1	21,08,8-2
3.1.2, 3.1.3	21,08,6
3.1.1	21,08,5
3.0.0	20,11,8

Argomenti

- [Configurazione di più code](#)
- [Slurmguida per la modalità a coda multipla](#)
- [Slurm modalità protetta dal cluster](#)
- [Slurmfailover rapido con capacità insufficiente del cluster](#)
- [Slurm pianificazione basata sulla memoria](#)
- [Allocazione di più tipi di istanza con Slurm](#)
- [Scalabilità del cluster per nodi dinamici](#)
- [Slurmcontabilità con AWS ParallelCluster](#)
- [Slurm personalizzazione della configurazione](#)
- [Slurm e prolog epilog](#)
- [Dimensioni e aggiornamento della capacità del cluster](#)

Configurazione di più code

Con AWS ParallelCluster la versione 3, è possibile configurare più code impostando `slurm` e specificando più di una coda nel file di [SlurmQueues](#) configurazione. [Scheduler](#) In questa modalità, coesistono diversi tipi di istanza nei nodi di calcolo specificati nella [ComputeResources](#) sezione del file di configurazione. [ComputeResources](#) con tipi di istanze diversi vengono ridimensionati verso l'alto o verso il basso in base alle esigenze di [SlurmQueues](#).

Le code multiple all'interno di un singolo cluster sono generalmente preferite rispetto a più cluster quando i carichi di lavoro condividono la stessa infrastruttura e le stesse risorse sottostanti (come storage condiviso, rete o nodi di accesso). Se i carichi di lavoro hanno esigenze di elaborazione, archiviazione e rete simili, l'utilizzo di più code all'interno di un singolo cluster è più efficiente perché consente la condivisione delle risorse ed evita inutili duplicazioni. Questo approccio semplifica la gestione e riduce il sovraccarico, pur consentendo una pianificazione efficiente dei lavori e l'allocazione delle risorse. D'altra parte, è necessario utilizzare più cluster quando vi sono forti requisiti di sicurezza, dati o isolamento operativo tra i carichi di lavoro. Ad esempio, se è necessario gestire e gestire i carichi di lavoro in modo indipendente, con pianificazioni, cicli di aggiornamento o politiche di accesso diversi, i cluster multipli sono più appropriati.

Quote di risorse di calcolo e coda dei cluster

Risorsa	Quota
Slurm queues	50 code per cluster
Compute resources	50 risorse di elaborazione per coda 50 risorse di elaborazione per cluster

Numero di nodi

Ogni risorsa di calcolo inserita in [ComputeResources](#) una coda deve avere un valore univoco [Name](#), [InstanceTypeMinCount](#), e. [MaxCount](#) [MinCount](#) e [MaxCount](#) hanno valori predefiniti che definiscono l'intervallo di istanze di una risorsa di calcolo in [ComputeResources](#) una coda. Puoi anche specificare i tuoi valori per e. [MinCountMaxCount](#) Ogni risorsa di elaborazione in [ComputeResources](#) è composta da nodi statici numerati da 1 al valore di [MinCount](#) e nodi dinamici numerati dal valore di [MinCount](#) al valore di. [MaxCount](#)

Configurazione di esempio

Di seguito è riportato un esempio di sezione [Scheduling](#) per un file di configurazione del cluster. In questa configurazione ci sono due code denominate queue1 queue2 e ciascuna di esse ha un valore specifico [ComputeResources](#). [MaxCount](#)

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - InstanceType: c5.xlarge
          MaxCount: 5
          Name: c5xlarge
        - InstanceType: c4.xlarge
          MaxCount: 5
          Name: c4xlarge
    - Name: queue2
      ComputeResources:
        - InstanceType: c5.xlarge
          MaxCount: 5
          Name: c5xlarge
```

Nomi host

Le istanze che vengono lanciate nel parco di elaborazione vengono assegnate dinamicamente. I nomi host vengono generati per ogni nodo. Per impostazione predefinita, AWS ParallelCluster utilizzerà il seguente formato del nome host:

```
$HOSTNAME=$QUEUE-$STATDYN-$COMPUTE_RESOURCE-$NODENUM
```

- `$QUEUE` è il nome della coda. Ad esempio, se la [SlurmQueues](#) sezione ha una voce `Name` impostata su «queue-name», allora «`$QUEUE`» è «queue-name».
- `$STATDYN` è `st` per nodi statici o `dy` per nodi dinamici.
- `$COMPUTE_RESOURCE` è la risorsa `Name` di [ComputeResources](#) calcolo corrispondente a questo nodo.
- `$NODENUM` è il numero del nodo. `$NODENUM` è compreso tra uno (1) e il valore di [MinCount](#) per i nodi statici e tra uno (1) e [MaxCount](#) - [MinCount](#) per i nodi dinamici.

Dal file di configurazione di esempio riportato sopra, un determinato nodo `queue1` e una risorsa di calcolo `c5xlarge` hanno un nome host: `queue1-dy-c5xlarge-1`

Sia i nomi host che i nomi di dominio completi (FQDN) vengono creati utilizzando le zone ospitate di Amazon Route 53. L'FQDN è `$HOSTNAME.$CLUSTERNAME.pcluster`, dov'è `$CLUSTERNAME` il nome del cluster.

Nota che lo stesso formato verrà utilizzato anche per i nomi dei Slurm nodi.

Gli utenti possono scegliere di utilizzare il EC2 nome host Amazon predefinito dell'istanza che alimenta il nodo di calcolo anziché il formato del nome host predefinito utilizzato da AWS ParallelCluster. Questo può essere fatto impostando il [UseEc2Hostnames](#) parametro su `true`. Tuttavia, i nomi dei Slurm nodi continueranno a utilizzare il AWS ParallelCluster formato predefinito.

Slurm guida per la modalità a coda multipla

Qui puoi scoprire come AWS ParallelCluster e Slurm gestire i nodi di coda (partizione) e come monitorare la coda e lo stato dei nodi.

Panoramica di

L'architettura di scalabilità si basa sulla [Cloud Scheduling Guide](#) e sul Slurm plug-in per il risparmio energetico. Per ulteriori informazioni sul plug-in per il risparmio energetico, consulta la Guida al

[risparmio Slurm energetico](#). Nell'architettura, le risorse che possono essere potenzialmente rese disponibili per un cluster sono in genere predefinite nella Slurm configurazione come nodi cloud.

Ciclo di vita dei nodi cloud

Durante il loro ciclo di vita, i nodi cloud entrano in diversi se non tutti i seguenti stati: **POWER_SAVING**, **POWER_UP** (`pow_up`), **IDLE** (`idle`) e **ALLOCATED** (`alloc`). **POWER_DOWN** (`pow_dn`) In alcuni casi, un nodo cloud potrebbe entrare nello **OFFLINE** stato. L'elenco seguente descrive in dettaglio diversi aspetti di questi stati nel ciclo di vita del nodo cloud.

- Un nodo in uno **POWER_SAVING** stato viene visualizzato con un `~` suffisso (ad esempio `idle~`) in. `sinfo` In questo stato, nessuna istanza EC2 esegue il backup del nodo. Tuttavia, Slurm può ancora allocare lavori al nodo.
- Un nodo in transizione verso uno **POWER_UP** stato viene visualizzato con un `#` suffisso (ad esempio `idle#`) in. `sinfo` Un nodo passa automaticamente a uno **POWER_UP** stato, quando Slurm assegna un lavoro a un nodo in uno stato. **POWER_SAVING**

In alternativa, puoi trasferire manualmente i nodi allo **POWER_UP** stato come utente su `root` con il comando:

```
$ scontrol update nodename=nodename state=power_up
```

In questa fase, `ResumeProgram` viene richiamato, le istanze EC2 vengono avviate e configurate e il nodo passa allo stato. **POWER_UP**

- Un nodo attualmente disponibile per l'uso viene visualizzato senza un suffisso (ad esempio) in. `idle` `sinfo` Dopo che il nodo è stato configurato ed è entrato a far parte del cluster, diventa disponibile per l'esecuzione dei job. In questa fase, il nodo è configurato correttamente e pronto per l'uso.

Come regola generale, consigliamo che il numero di istanze Amazon EC2 sia uguale al numero di nodi disponibili. Nella maggior parte dei casi, i nodi statici sono disponibili dopo la creazione del cluster.

- Un nodo che sta passando a **POWER_DOWN** uno stato viene visualizzato con un `%` suffisso (ad esempio `idle%`) in. `sinfo` I nodi dinamici entrano automaticamente nello **POWER_DOWN** stato successivo. [ScaledownIdletime](#) Al contrario, i nodi statici nella maggior parte dei casi non vengono spenti. Tuttavia, puoi posizionare i nodi nello **POWER_DOWN** stato manualmente come utente su `root` con il comando:

```
$ scontrol update nodename=nodename state=down reason="manual draining"
```

In questo stato, le istanze associate a un nodo vengono terminate e il nodo viene ripristinato allo POWER_SAVING stato e può essere utilizzato successivamente. [ScaledownIdleTime](#)

L'[ScaledownIdleTime](#) impostazione viene salvata nell'impostazione di Slurm configurazione `SuspendTimeout`.

- Viene visualizzato un nodo offline con un * suffisso (ad esempio `down*`) in `sinfo`. Un nodo va offline se il Slurm controller non riesce a contattare il nodo o se i nodi statici sono disabilitati e le istanze di backup vengono terminate.

Considerate gli stati dei nodi mostrati nell'esempio seguente. `sinfo`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite    4  idle~ efa-dy-efacompute1-[1-4]
efa        up    infinite    1  idle  efa-st-efacompute1-1
gpu        up    infinite    1  idle% gpu-dy-gpucompute1-1
gpu        up    infinite    9  idle~ gpu-dy-gpucompute1-[2-10]
ondemand   up    infinite    2  mix#  ondemand-dy-ondemandcompute1-[1-2]
ondemand   up    infinite   18  idle~ ondemand-dy-ondemandcompute1-
[3-10],ondemand-dy-ondemandcompute2-[1-10]
spot*      up    infinite   13  idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*      up    infinite    2  idle  spot-st-spotcompute2-[1-2]
```

I `efa-st-efacompute1-1` nodi `spot-st-spotcompute2-[1-2]` and dispongono già di istanze di backup configurate e sono disponibili per l'uso. I `ondemand-dy-ondemandcompute1-[1-2]` nodi sono nello POWER_UP stato attuale e dovrebbero essere disponibili entro pochi minuti. Il `gpu-dy-gpucompute1-1` nodo è nello POWER_DOWN stato e passa allo POWER_SAVING stato successivo [ScaledownIdleTime](#) (il valore predefinito è 10 minuti).

Tutti gli altri nodi sono in POWER_SAVING stato e non sono supportati da istanze EC2.

Lavorare con un nodo disponibile

Un nodo disponibile è supportato da un'istanza Amazon EC2. Per impostazione predefinita, il nome del nodo può essere utilizzato per inserire direttamente SSH nell'istanza (ad esempio `ssh efa-st-efacompute1-1`). L'indirizzo IP privato dell'istanza può essere recuperato utilizzando il comando:

```
$ scontrol show nodes nodename
```

Controlla l'indirizzo IP nel NodeAddr campo restituito.

Per i nodi che non sono disponibili, il NodeAddr campo non deve puntare a un'istanza Amazon EC2 in esecuzione. Piuttosto, dovrebbe essere lo stesso del nome del nodo.

Job stati e invio

I lavori inviati nella maggior parte dei casi vengono immediatamente assegnati ai nodi del sistema o messi in sospeso se tutti i nodi sono allocati.

Se i nodi allocati per un processo includono nodi in uno POWER_SAVING stato, il processo inizia con uno CF stato o. CONFIGURING A questo punto, il processo attende che i nodi dello POWER_SAVING stato passino allo POWER_UP stato e diventino disponibili.

Dopo che tutti i nodi allocati per un lavoro sono disponibili, il lavoro entra nello stato RUNNING (R).

Per impostazione predefinita, tutti i lavori vengono inviati alla coda predefinita (nota come partizione in). Slurm Ciò è indicato da un * suffisso dopo il nome della coda. È possibile selezionare una coda utilizzando l'opzione di invio del -p lavoro.

Tutti i nodi sono configurati con le seguenti funzionalità, che possono essere utilizzate nei comandi di invio dei lavori:

- Un tipo di istanza (ad esempio `c5.xlarge`)
- Un tipo di nodo (questo è `dynamic` o `static`.)

Puoi vedere le caratteristiche di un particolare nodo usando il comando:

```
$ scontrol show nodes nodename
```

Nel ritorno, controlla l'AvailableFeatureselenco.

Considera lo stato iniziale del cluster, che puoi visualizzare eseguendo il `sinfo` comando.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite     4  idle~ efa-dy-efacompute1-[1-4]
efa        up    infinite     1  idle efa-st-efacompute1-1
```



```

gpu          up    infinite    10  idle~ gpu-dy-gpucompute1-[1-10]
ondemand     up    infinite    20  idle~ ondemand-dy-ondemandcompute1-
[1-10],ondemand-dy-ondemandcompute2-[1-10]
spot*        up    infinite    13  idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*        up    infinite     2  idle spot-st-spotcompute2-[1-2]

```

Nota che spot è la coda predefinita. È indicata dal * suffisso.

Invia un lavoro a un nodo statico nella coda predefinita ()spot.

```
$ sbatch --wrap "sleep 300" -N 1 -C static
```

Invia un lavoro a un nodo dinamico nella EFA coda.

```
$ sbatch --wrap "sleep 300" -p efa -C dynamic
```

Invia un lavoro a otto (8) c5.2xlarge nodi e due (2) t2.xlarge nodi nella ondemand coda.

```
$ sbatch --wrap "sleep 300" -p ondemand -N 10 -C "[c5.2xlarge*8&t2.xlarge*2]"
```

Invia un lavoro a un nodo GPU nella gpu coda.

```
$ sbatch --wrap "sleep 300" -p gpu -G 1
```

Considerate lo stato dei lavori che utilizzano il squeue comando.

```

$ squeue
JOBID PARTITION  NAME  USER  ST      TIME  NODES NODELIST(REASON)
  12  ondemand    wrap  ubuntu CF      0:36     10 ondemand-dy-ondemandcompute1-
[1-8],ondemand-dy-ondemandcompute2-[1-2]
  13      gpu    wrap  ubuntu CF      0:05      1 gpu-dy-gpucompute1-1
   7      spot    wrap  ubuntu R      2:48      1 spot-st-spotcompute2-1
   8      efa    wrap  ubuntu R      0:39      1 efa-dy-efacompute1-1

```

I lavori 7 e 8 (nelle efa code spot e) sono già in esecuzione (R). I lavori 12 e 13 sono ancora in fase di configurazione (CF), probabilmente in attesa che le istanze diventino disponibili.

```

# Nodes states corresponds to state of running jobs
$ sinfo

```

PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
efa	up	infinite	3	idle~	efa-dy-efacompute1-[2-4]
efa	up	infinite	1	mix	efa-dy-efacompute1-1
efa	up	infinite	1	idle	efa-st-efacompute1-1
gpu	up	infinite	1	mix~	gpu-dy-gpucompute1-1
gpu	up	infinite	9	idle~	gpu-dy-gpucompute1-[2-10]
ondemand	up	infinite	10	mix#	ondemand-dy-ondemandcompute1-[1-8],ondemand-dy-ondemandcompute2-[1-2]
ondemand	up	infinite	10	idle~	ondemand-dy-ondemandcompute1-[9-10],ondemand-dy-ondemandcompute2-[3-10]
spot*	up	infinite	13	idle~	spot-dy-spotcompute1-[1-10],spot-dy-spotcompute2-[1-3]
spot*	up	infinite	1	mix	spot-st-spotcompute2-1
spot*	up	infinite	1	idle	spot-st-spotcompute2-2

Stato e caratteristiche del nodo

Nella maggior parte dei casi, gli stati dei nodi sono completamente gestiti in AWS ParallelCluster base ai processi specifici del ciclo di vita dei nodi cloud descritti in precedenza in questo argomento.

Tuttavia, sostituisce o termina AWS ParallelCluster anche nodi non integri in DRAINED stati DOWN e nodi con istanze di backup non integre. Per ulteriori informazioni, consulta [clustermgtd](#).

Stati di partizione

AWS ParallelCluster supporta i seguenti stati di partizione. Una Slurm partizione è una coda in entrata. AWS ParallelCluster

- UP: indica che la partizione è in uno stato attivo. Questo è lo stato predefinito di una partizione. In questo stato, tutti i nodi della partizione sono attivi e disponibili per l'uso.
- INACTIVE: indica che la partizione è inattiva. In questo stato, tutte le istanze che supportano i nodi di backup di una partizione inattiva vengono terminate. Non vengono avviate nuove istanze per i nodi in una partizione inattiva.

pcluster update-compute-fleet

- Arresto del parco di calcolo: quando viene eseguito il comando seguente, tutte le partizioni passano INACTIVE allo stato e i AWS ParallelCluster processi mantengono le partizioni nello stato. INACTIVE

```
$ pcluster update-compute-fleet --cluster-name testSlurm \
```

```
--region eu-west-1 --status STOP_REQUESTED
```

- Avvio del parco di calcolo: quando viene eseguito il comando seguente, tutte le partizioni passano inizialmente allo stato. UP Tuttavia, AWS ParallelCluster i processi non mantengono la partizione in uno stato. UP È necessario modificare manualmente lo stato delle partizioni. Tutti i nodi statici diventano disponibili dopo pochi minuti. Tieni presente che l'impostazione di una partizione su UP non attiva alcuna capacità dinamica.

```
$ pcluster update-compute-fleet --cluster-name testSlurm \  
  --region eu-west-1 --status START_REQUESTED
```

Quando `update-compute-fleet` viene eseguito, è possibile verificare lo stato del cluster eseguendo il `pcluster describe-compute-fleet` comando e controllando il `Status`. Di seguito sono elencati gli stati possibili:

- **STOP_REQUESTED**: La richiesta di stop compute fleet viene inviata al cluster.
- **STOPPING**: Il `pcluster` processo sta attualmente interrompendo la flotta di elaborazione.
- **STOPPED**: Il `pcluster` processo ha terminato il processo di arresto, tutte le partizioni sono in **INACTIVE** stato attivo e tutte le istanze di calcolo sono terminate.
- **START_REQUESTED**: La richiesta di avvio della flotta di calcolo viene inviata al cluster.
- **STARTING**: Il `pcluster` processo sta attualmente avviando il cluster.
- **RUNNING**: Il `pcluster` processo ha completato il processo di avvio, tutte le partizioni sono nello UP stato attuale e i nodi statici sono disponibili dopo alcuni minuti.
- **PROTECTED**: Questo stato indica che alcune partizioni presentano errori di bootstrap consistenti. Le partizioni interessate sono inattive. Esamina il problema e poi corri `update-compute-fleet` a riattivare la flotta.

Controllo manuale delle code

In alcuni casi, potresti voler avere un controllo manuale sui nodi o sulla coda (nota come partizione in Slurm) in un cluster. È possibile gestire i nodi in un cluster tramite le seguenti procedure comuni utilizzando il `scontrol` comando.

- Accendi i nodi dinamici in **POWER_SAVING** stato

Esegui il comando come utente su root:

```
$ scontrol update nodename=nodename state=power_up
```

Puoi anche inviare un `sleep 1` lavoro segnaposto richiedendo un certo numero di nodi e poi fare affidamento su di esso Slurm per attivare il numero richiesto di nodi.

- Spegni prima i nodi dinamici [ScaledownIdletime](#)

Ti consigliamo di impostare i nodi dinamici DOWN come utente su root con il comando:

```
$ scontrol update nodename=nodename state=down reason="manually draining"
```

AWS ParallelCluster termina e ripristina automaticamente i nodi dinamici disattivati.

In generale, non è consigliabile impostare i nodi in modo che utilizzino `POWER_DOWN` direttamente il comando. `scontrol update nodename=nodename state=power_down` Questo perché gestisce AWS ParallelCluster automaticamente il processo di spegnimento.

- Disabilita una coda (partizione) o ferma tutti i nodi statici in una partizione specifica

Imposta una coda specifica INACTIVE come utente su root con il comando:

```
$ scontrol update partition=queuename state=inactive
```

Questa operazione interrompe tutte le istanze che eseguono il backup dei nodi nella partizione.

- Abilita una coda (partizione)

Imposta una coda specifica per UP un utente su root con il comando:

```
$ scontrol update partition=queuename state=up
```

Comportamento e regolazioni di ridimensionamento

Ecco un esempio del normale flusso di lavoro di ridimensionamento:

- Lo scheduler riceve un lavoro che richiede due nodi.
- Lo scheduler trasferisce due nodi in uno `POWER_UP` stato e chiama `ResumeProgram` con i nomi dei nodi (ad esempio). `queue1-dy-spotcompute1-[1-2]`

- `ResumeProgram` avvia due istanze Amazon EC2 e assegna gli indirizzi IP e i nomi host privati di, in `ResumeTimeout` attesa (il periodo predefinito è `queue1-dy-spotcompute1-[1-2]` di 30 minuti) prima di reimpostare i nodi.
- Le istanze vengono configurate e si uniscono al cluster. Un processo inizia a essere eseguito sulle istanze.
- Il processo viene completato e l'esecuzione si interrompe.
- Al termine della configurazione `SuspendTime` (che è impostata su [ScaledownIdleTime](#)), lo scheduler imposta le istanze sullo stato. `POWER_SAVING` Lo scheduler imposta quindi `POWER_DOWN` lo stato e chiama `queue1-dy-spotcompute1-[1-2]` `SuspendProgram` con i nomi dei nodi.
- `SuspendProgram` viene chiamato per due nodi. I nodi rimangono nello `POWER_DOWN` stato, ad esempio, rimanendo `idle%` per `a SuspendTimeout` (il periodo predefinito è 120 secondi (2 minuti)). Dopo aver `clustermgtd` rilevato che i nodi si stanno spegnendo, interrompe le istanze di backup. Quindi, passa `queue1-dy-spotcompute1-[1-2]` allo stato di inattività e reimposta l'indirizzo IP privato e il nome host in modo che sia pronto per l'avvio per i lavori futuri.

Se qualcosa va storto e un'istanza per un particolare nodo non può essere avviata per qualche motivo, succede quanto segue:

- Lo scheduler riceve un lavoro che richiede due nodi.
- Lo scheduler trasferisce due nodi di cloud bursting `POWER_UP` allo stato e chiama `ResumeProgram` con i nomi dei nodi, (ad esempio). `queue1-dy-spotcompute1-[1-2]`
- `ResumeProgram` avvia solo una (1) istanza Amazon EC2 e `queue1-dy-spotcompute1-1` configura, con una (1) istanza `queue1-dy-spotcompute1-2`, che non riesce ad avviarsi.
- `queue1-dy-spotcompute1-1` non è interessato e torna online dopo aver raggiunto lo stato. `POWER_UP`
- `queue1-dy-spotcompute1-2` passa allo `POWER_DOWN` stato e il processo viene richiesto automaticamente perché Slurm rileva un errore del nodo.
- `queue1-dy-spotcompute1-2` diventa disponibile dopo `SuspendTimeout` (l'impostazione predefinita è 120 secondi (2 minuti)). Nel frattempo, il processo viene richiesto e può iniziare a essere eseguito su un altro nodo.
- Il processo precedente si ripete finché il processo non può essere eseguito su un nodo disponibile senza che si verifichi un errore.

Esistono due parametri di temporizzazione che possono essere regolati se necessario:

- **ResumeTimeout**(l'impostazione predefinita è 30 minuti): ResumeTimeout controlla il tempo di Slurm attesa prima della transizione del nodo allo stato inattivo.
 - Potrebbe essere utile estenderlo ResumeTimeout se il processo di pre/post installazione richiede quasi così tanto tempo.
 - ResumeTimeout è anche il tempo massimo di AWS ParallelCluster attesa prima di sostituire o resettare un nodo in caso di problemi. I nodi di calcolo si interrompono automaticamente se si verifica un errore durante l'avvio o la configurazione. AWS ParallelCluster i processi sostituiscono un nodo al rilevamento di un'istanza terminata.
- **SuspendTimeout**(l'impostazione predefinita è 120 secondi (2 minuti)): SuspendTimeout controlla la velocità con cui i nodi vengono reinseriti nel sistema e sono nuovamente pronti per l'uso.
 - Un valore più corto SuspendTimeout indica che i nodi vengono ripristinati più rapidamente e Slurm possono provare ad avviare le istanze più frequentemente.
 - Un valore più lungo SuspendTimeout significa che i nodi guasti vengono ripristinati più lentamente. Nel frattempo, Slurm prova a usare altri nodi. Se SuspendTimeout dura più di qualche minuto, Slurm prova a scorrere ciclicamente tra tutti i nodi del sistema. Una versione più lunga SuspendTimeout potrebbe essere utile per i sistemi su larga scala (oltre 1.000 nodi) per ridurre lo stress Slurm quando si tenta di rimettere spesso in coda i lavori non riusciti.
 - Tieni presente che SuspendTimeout non si riferisce al tempo di AWS ParallelCluster attesa per terminare un'istanza di backup per un nodo. Le istanze di backup per POWER_DOWN i nodi vengono immediatamente terminate. Il processo di terminazione di solito termina in pochi minuti. Tuttavia, durante questo periodo, il nodo rimane nello POWER_DOWN stato e non è disponibile per l'uso da parte dello scheduler.

Registri per l'architettura

L'elenco seguente contiene i log delle chiavi. Il nome del flusso di log utilizzato con Amazon CloudWatch Logs ha il formato `{hostname}.{instance_id}.{logIdentifier}`
logIdentifier seguente i nomi di log.

- ResumeProgram: /var/log/parallelcluster/slurm_resume.log (slurm_resume)
- SuspendProgram: /var/log/parallelcluster/slurm_suspend.log (slurm_suspend)
- clustermgtd: /var/log/parallelcluster/clustermgtd.log (clustermgtd)

- `computemgtd`: `/var/log/parallelcluster/computemgtd.log` (`computemgtd`)
- `slurmctld`: `/var/log/slurmctld.log` (`slurmctld`)
- `slurmd`: `/var/log/slurmd.log` (`slurmd`)

Problemi comuni e modalità di debug:

Nodi che non sono riusciti ad avviarsi, accendersi o unirsi al cluster

- Nodi dinamici:
 - Controlla il `ResumeProgram` registro per vedere se `ResumeProgram` è stato chiamato con il nodo. In caso contrario, controlla il `slurmctld` registro per determinare se hai Slurm provato a chiamare `ResumeProgram` con il nodo. Tieni presente che autorizzazioni errate `ResumeProgram` potrebbero causare l'interruzione automatica del programma.
 - Se `ResumeProgram` viene chiamato, controlla se è stata lanciata un'istanza per il nodo. Se l'istanza non è stata avviata, dovrebbe apparire un messaggio di errore chiaro sul motivo per cui l'istanza non è stata avviata.
 - Se è stata avviata un'istanza, potrebbe essersi verificato qualche problema durante il processo di bootstrap. Trova l'indirizzo IP privato e l'ID dell'istanza corrispondenti dal `ResumeProgram` registro e guarda i registri di bootstrap corrispondenti per l'istanza specifica in Logs. CloudWatch
- Nodi statici:
 - Controlla il `clustermgtd` registro per vedere se sono state avviate istanze per il nodo. Se le istanze non sono state avviate, dovrebbero esserci errori evidenti sul motivo per cui le istanze non sono state avviate.
 - Se è stata avviata un'istanza, c'è qualche problema con il processo di bootstrap. Trova l'IP privato e l'ID dell'istanza corrispondenti dal `clustermgtd` registro e guarda i registri di bootstrap corrispondenti per l'istanza specifica in Logs. CloudWatch

Nodi sostituiti o terminati in modo imprevisto e guasti dei nodi

- `replaced/terminated` Nodi in modo imprevisto:
 - Nella maggior parte dei casi, `clustermgtd` gestisce tutte le azioni di manutenzione dei nodi. Per verificare se un nodo è stato `clustermgtd` sostituito o interrotto, controlla il `clustermgtd` registro.
 - Se il nodo è stato `clustermgtd` sostituito o terminato, dovrebbe apparire un messaggio che indica il motivo dell'azione. Se il motivo è legato allo scheduler (ad esempio, il nodo `lo eraDOWN`),

controlla il `slurmctld` registro per maggiori dettagli. Se il motivo è correlato ad Amazon EC2, utilizza strumenti come Amazon CloudWatch o la console Amazon EC2, l'interfaccia a riga di comando o gli SDK per verificare lo stato o i log di quell'istanza. Ad esempio, puoi verificare se l'istanza aveva eventi pianificati o non ha superato i controlli dello stato di integrità di Amazon EC2.

- Se `clustermgtd` non ha terminato il nodo, controlla se ha `computemgtd` terminato il nodo o se EC2 ha terminato l'istanza per recuperare un'istanza Spot.
- Guasti ai nodi:
 - Nella maggior parte dei casi, i lavori vengono richiesti automaticamente in caso di errore di un nodo. Esamina nel `slurmctld` registro il motivo per cui un job o un nodo non è riuscito e valuta la situazione da lì.

Guasto durante la sostituzione o la chiusura delle istanze, errore durante lo spegnimento dei nodi

- In generale, `clustermgtd` gestisce tutte le azioni di terminazione previste dell'istanza. Guarda nel `clustermgtd` registro per vedere perché non è riuscito a sostituire o terminare un nodo.
- Se i nodi dinamici non funzionano [ScaledownIdleTime](#) correttamente, guarda nel `SuspendProgram` registro per vedere se `slurmctld` i processi hanno effettuato chiamate con il nodo specifico come argomento. Note in realtà `SuspendProgram` non esegue alcuna azione specifica. Piuttosto, registra solo quando viene chiamato. Tutte le terminazioni e i `NodeAddr` ripristini delle istanze vengono completati da `clustermgtd`. `Slurm` trasferisce i nodi a dopo. `IDLE SuspendTimeout`

Altri problemi:

- AWS ParallelCluster non prende decisioni sull'allocazione del lavoro o sulla scalabilità. Tenta solo di avviare, terminare e mantenere le risorse in base alle `Slurm` istruzioni fornite.

Per problemi relativi all'allocazione dei lavori, all'allocazione dei nodi e alla decisione sulla scalabilità, consulta il `slurmctld` registro per individuare eventuali errori.

Slurm modalità protetta dal cluster

Quando un cluster viene eseguito con la modalità protetta abilitata, AWS ParallelCluster monitora e tiene traccia degli errori di bootstrap dei nodi di calcolo durante l'avvio dei nodi di calcolo. Lo fa per rilevare se questi errori si verificano continuamente.

Se in una coda (partizione) viene rilevato quanto segue, il cluster entra nello stato protetto:

1. Gli errori consecutivi di bootstrap dei nodi di calcolo si verificano continuamente senza che i nodi di calcolo vengano avviati correttamente.
2. Il numero di errori raggiunge una soglia predefinita.

Dopo che il cluster ha raggiunto lo stato protetto, AWS ParallelCluster disabilita le code con errori pari o superiori alla soglia predefinita.

Slurm la modalità protetta dal cluster è stata aggiunta nella versione 3.0.0. AWS ParallelCluster

È possibile utilizzare la modalità protetta per ridurre il tempo e le risorse impiegate per il ciclo di errore di avvio del nodo di calcolo.

Parametro della modalità protetta

protected_failure_count

`protected_failure_count` specifica il numero di errori consecutivi in una coda (partizione) che attivano lo stato di protezione del cluster.

L'impostazione predefinita `protected_failure_count` è 10 e la modalità protetta è abilitata.

Se `protected_failure_count` è maggiore di zero, la modalità protetta è abilitata.

Se `protected_failure_count` è minore o uguale a zero, la modalità protetta è disabilitata.

È possibile modificare il `protected_failure_count` valore aggiungendo il parametro nel file di `clustermgtd` configurazione che si trova `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` in. `HeadNode`

Puoi aggiornare questo parametro in qualsiasi momento e non è necessario interrompere la flotta di elaborazione per farlo. Se un avvio riesce in una coda prima del raggiungimento del numero di errori `protected_failure_count`, il conteggio degli errori viene azzerato.

Controlla lo stato del cluster in stato protetto

Quando un cluster è in stato protetto, puoi controllare lo stato del parco di elaborazione e gli stati dei nodi.

Elaborazione dello stato del parco veicoli

Lo stato della flotta di elaborazione si trova `PROTECTED` in un cluster in esecuzione in stato protetto.

```
$ pcluster describe-compute-fleet --cluster-name <cluster-name> --region <region-id>
{
  "status": "PROTECTED",
  "lastStatusUpdateTime": "2022-04-22T00:31:24.000Z"
}
```

Stato del nodo

Per sapere quali code (partizioni) presentano errori di bootstrap che hanno attivato lo stato protetto, accedi al cluster ed esegui il comando. `sinfo` Le partizioni con errori di bootstrap pari o superiori sono nello stato. `protected_failure_count` INACTIVE Le partizioni senza errori di bootstrap pari o superiori `protected_failure_count` sono nello stato e funzionano come previsto. UP

PROTECTED lo stato non influisce sull'esecuzione dei lavori. Se i job sono in esecuzione su una partizione con errori di bootstrap pari o superiori `protected_failure_count`, la partizione viene impostata INACTIVE dopo il completamento dei job in esecuzione.

Considerate gli stati dei nodi mostrati nell'esempio seguente.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* inact infinite 10 down% queue1-dy-c5xlarge-[1-10]
queue1* inact infinite 3490 idle~ queue1-dy-c5xlarge-[11-3500]
queue2 up infinite 10 idle~ queue2-dy-c5xlarge-[1-10]
```

queue1 La partizione è INACTIVE dovuta al fatto che sono stati rilevati 10 errori di bootstrap consecutivi dei nodi di calcolo.

Le istanze relative ai nodi sono state `queue1-dy-c5xlarge-[1-10]` avviate ma non sono riuscite a unirsi al cluster a causa di uno stato non integro.

Lo stato del cluster è protetto.

La partizione `queue2` non è influenzata dagli errori di bootstrap in. `queue1` È nello UP stato e può ancora eseguire lavori.

Come disattivare lo stato protetto

Dopo aver risolto l'errore di bootstrap, puoi eseguire il seguente comando per disattivare lo stato di protezione del cluster.

```
$ pcluster update-compute-fleet --cluster-name <cluster-name> \  
  --region <region-id> \  
  --status START_REQUESTED
```

Errori di bootstrap che attivano lo stato protetto

Gli errori di bootstrap che attivano lo stato protetto sono suddivisi nei tre tipi seguenti. Per identificare il tipo e il problema, puoi verificare se i log sono stati AWS ParallelCluster generati. Se i log sono stati generati, puoi controllarli per i dettagli degli errori. Per ulteriori informazioni, consulta [Recupero e conservazione dei log](#).

1. Errore di bootstrap che causa la terminazione automatica di un'istanza.

Un'istanza ha esito negativo nelle prime fasi del processo di bootstrap, ad esempio un'istanza che si interrompe automaticamente a causa di errori nello script\|.

[SlurmQueuesCustomActionsOnNodeStartOnNodeConfigured](#)

Per i nodi dinamici, cercate errori simili ai seguenti:

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

Per i nodi statici, cerca nel `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) errori simili ai seguenti:

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

2. Nodi **resume_timeout** o **node_replacement_timeout** scadenze.

Un'istanza non può entrare a far parte del cluster all'interno di `resume_timeout` (per nodi dinamici) o `node_replacement_timeout` (per nodi statici). Non si interrompe automaticamente prima del timeout. Ad esempio, la rete non è configurata correttamente per il cluster e il nodo è impostato sullo stato da DOWN Slurm dopo la scadenza del timeout.

Per i nodi dinamici, cerca errori simili ai seguenti:

```
Node bootstrap error: Resume timeout expires for node
```

Per i nodi statici, cerca nel `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) errori simili ai seguenti:

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

3. I nodi non superano il controllo di integrità.

Un'istanza dietro il nodo non supera il controllo dello stato di EC2 integrità di Amazon o il controllo dello stato di un evento pianificato e i nodi vengono trattati come nodi di errore di bootstrap. In questo caso, l'istanza si interrompe per un motivo che sfugge al controllo di. AWS ParallelCluster

Cerca nel `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) gli errori simili ai seguenti:

```
Node bootstrap error: Node %s failed during bootstrap when performing health check.
```

4. I nodi di calcolo falliscono Slurm registrazione.

La registrazione del `slurmd` demone con Slurm control daemon (`slurmctld`) fallisce e fa sì che lo stato del nodo di calcolo passi allo stato. `INVALID_REG` Configurato in modo errato Slurm i nodi di calcolo possono causare questo errore, ad esempio i nodi calcolati configurati con errori di specificazione dei nodi di [CustomSlurmSettings](#) calcolo.

Cerca nel file di `slurmctld` log (`/var/log/slurmctld.log`) sul nodo principale o cerca nel file di `slurmd` log (`/var/log/slurmd.log`) del nodo di calcolo fallito gli errori simili ai seguenti:

```
Setting node %s to INVALID with reason: ...
```

Come eseguire il debug della modalità protetta

Se lo stato del cluster è protetto e se `clustermgtd` i log sono AWS ParallelCluster stati generati da HeadNode e i `cloud-init-output` log dai nodi di calcolo problematici, puoi controllare i log per i dettagli sugli errori. Per ulteriori informazioni su come recuperare i log, consulta. [Recupero e conservazione dei log](#)

`clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) sul nodo principale

I messaggi di registro mostrano quali partizioni presentano errori di bootstrap e il numero di errori di bootstrap corrispondente.

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - INFO - Partitions
```

```
bootstrap failure count: {'queue1': 2}, cluster will be set into protected mode if
protected failure count reach threshold.
```

Nel `clustermgtd` registro, cerca per trovare quale nodo non è Found the following bootstrap failure nodes riuscito a eseguire il bootstrap.

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - WARNING -
Found the following bootstrap failure nodes: (x2) ['queue1-st-
c5large-1(192.168.110.155)', 'broken-st-c5large-2(192.168.65.215)']
```

Nel `clustermgtd` registro, cerca per Node bootstrap error trovare il motivo dell'errore.

```
[slurm_plugin.clustermgtd:_is_node_bootstrap_failure] - WARNING - Node bootstrap
error:
Node broken-st-c5large-2(192.168.65.215) is currently in replacement and no backing
instance
```

cloud-init-outputlog (/var/log/cloud-init-output.log) sui nodi di calcolo

Dopo aver ottenuto l'indirizzo IP privato del nodo di errore di avvio nel `clustermgtd` registro, puoi trovare il registro del nodo di calcolo corrispondente accedendo al nodo di calcolo o seguendo le istruzioni per recuperare i log. [Recupero e conservazione dei log](#) Nella maggior parte dei casi, il `/var/log/cloud-init-output` log del nodo problematico mostra il passaggio che ha causato l'errore di bootstrap del nodo di calcolo.

Slurmfailover rapido con capacità insufficiente del cluster

A partire dalla AWS ParallelCluster versione 3.2.0, i cluster vengono eseguiti con la modalità di failover rapido con capacità insufficiente abilitata per impostazione predefinita. Ciò riduce al minimo il tempo impiegato per riprovare a mettere in coda un lavoro quando vengono rilevati errori di capacità insufficiente di Amazon EC2. Ciò è particolarmente efficace quando si configura la coda con più risorse di elaborazione che utilizzano tipi di istanze diversi.

Amazon EC2 ha rilevato errori di capacità insufficienti:

- `InsufficientInstanceCapacity`
- `InsufficientHostCapacity`
- `InsufficientReservedInstanceCapacity`

- `MaxSpotInstanceCountExceeded`
- `SpotMaxPriceTooLow`: attivato se il prezzo della richiesta Spot è inferiore al prezzo minimo di gestione delle richieste Spot richiesto.
- `Unsupported`: Attivato con l'uso di un tipo di istanza che non è supportato in uno specifico. Regione AWS

In modalità di failure-over rapido con capacità insufficiente, se viene rilevato un errore di capacità insufficiente quando un lavoro viene assegnato a [SlurmQueues/compute_resource](#), AWS ParallelCluster effettua le seguenti operazioni:

1. Imposta la risorsa di calcolo su uno stato disabilitato (DOWN) per un periodo di tempo predefinito.
2. Viene utilizzato `POWER_DOWN_FORCE` per annullare i job del nodo in errore della risorsa di calcolo e per sospendere il nodo in errore. Imposta il nodo in errore `POWER_DOWN (!)` sullo stato `IDLE` and e quindi su `POWERING_DOWN (%)`
3. Richiede il lavoro a un'altra risorsa di elaborazione.

I nodi statici e accesi della risorsa di elaborazione disattivata non sono interessati. I lavori possono essere completati su questi nodi.

Questo ciclo si ripete finché il lavoro non viene assegnato correttamente a uno o più nodi di risorse di elaborazione. Per informazioni sugli stati dei nodi, vedere [Slurmguida per la modalità a coda multipla](#)

Se non viene trovata alcuna risorsa di elaborazione per eseguire il processo, il processo viene impostato `PENDING` sullo stato fino allo scadere del periodo di tempo predefinito. In questo caso, è possibile modificare il periodo di tempo predefinito come descritto nella sezione seguente.

Parametro di timeout della capacità insufficiente

`insufficient_capacity_timeout`

`insufficient_capacity_timeout` specifica il periodo di tempo (in secondi) in cui la risorsa di elaborazione viene mantenuta nello stato disabled (down) quando viene rilevato un errore di capacità insufficiente.

Per impostazione predefinita, `insufficient_capacity_timeout` è abilitato.

L'impostazione predefinita `insufficient_capacity_timeout` è 600 secondi (10 minuti).

Se il `insufficient_capacity_timeout` valore è inferiore o uguale a zero, la modalità di failure-over rapido della capacità insufficiente viene disattivata.

È possibile modificare il `insufficient_capacity_timeout` valore aggiungendo il parametro nel file di `clustermgtd` configurazione che si trova in `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` HeadNode

Il parametro può essere aggiornato in qualsiasi momento senza interrompere il parco di elaborazione.

Esempio:

- `insufficient_capacity_timeout=600:`

Se viene rilevato un errore di capacità insufficiente, la risorsa di calcolo viene impostata su `disabled ()`. `DOWN` Dopo 10 minuti, il relativo nodo guasto viene impostato sullo stato `idle~` (`POWER_SAVING`).

- `insufficient_capacity_timeout=60:`

Se viene rilevato un errore di capacità insufficiente, la risorsa di calcolo è disattivata (`DOWN`). Dopo 1 minuto, il nodo guasto viene impostato sullo `idle~` stato.

- `insufficient_capacity_timeout=0:`

La modalità di failure-over rapido della capacità insufficiente è disattivata. La risorsa di calcolo non è disabilitata.

Note

Potrebbe verificarsi un ritardo fino a un minuto tra il momento in cui i nodi si guastano a causa di errori di capacità insufficiente e il momento in cui il daemon di gestione del cluster rileva i guasti dei nodi. Questo perché il daemon di gestione del cluster verifica la presenza di errori di capacità insufficiente dei nodi e imposta lo stato delle risorse di calcolo a intervalli di un minuto. `down`

Stato rapido della modalità di failover con capacità insufficiente

Quando un cluster è in modalità di failover rapido con capacità insufficiente, è possibile verificarne lo stato e lo stato del nodo.

Stati dei nodi

Quando un lavoro viene inviato a un nodo dinamico di risorse di calcolo e viene rilevato un errore di capacità insufficiente, il nodo viene posizionato nello `down#` stato corrispondente.

```
(Code:InsufficientInstanceCapacity)Failure when resuming nodes.
```

Quindi i nodi spenti (nodi in `idle~` stato) vengono impostati `down~` con motivo.

```
(Code:InsufficientInstanceCapacity)Temporarily disabling node due to insufficient capacity.
```

Il lavoro viene richiesto ad altre risorse di elaborazione in coda.

I nodi statici delle risorse di elaborazione e i nodi che UP non sono interessati dalla modalità di failover rapida con capacità insufficiente.

Considerate gli stati dei nodi mostrati nell'esempio seguente.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite    30   idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up      infinite    30   idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

Inoltriamo un lavoro a `queue1` che richiede un nodo.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite    1   down# queue1-dy-c-1-1
queue1*   up      infinite    15  idle~ queue1-dy-c-2-[1-15]
queue1*   up      infinite    14  down~ queue1-dy-c-1-[2-15]
queue2    up      infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

`queue1-dy-c-1-1` Il nodo viene avviato per eseguire il lavoro. Tuttavia, l'istanza non è stata avviata a causa di un errore di capacità insufficiente. `queue1-dy-c-1-1` Il nodo è impostato su `down`. Il nodo dinamico spento all'interno della risorsa di calcolo (`queue2-dy-c-1`) è impostato su `down`.

Puoi controllare il motivo del nodo con `scontrol show nodes`.

```
$ scontrol show nodes queue1-dy-c-1-1
```



```

NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Failure when resuming nodes
[root@2022-03-10T22:17:50]

$ scontrol show nodes queue1-dy-c-1-2
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Temporarily disabling node due to
insufficient capacity [root@2022-03-10T22:17:50]

```

Il lavoro viene messo in coda su un altro tipo di istanza all'interno delle risorse di calcolo della coda.

Al termine, `insufficient_capacity_timeout` i nodi della risorsa di calcolo vengono ripristinati allo stato. `idle~`

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite    30   idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up      infinite    30   idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]

```

Dopo che gli `insufficient_capacity_timeout` intervalli e i nodi della risorsa di calcolo sono stati ripristinati `idle~` allo stato, lo Slurm scheduler assegna ai nodi una priorità inferiore. Lo scheduler continua a selezionare nodi da altre risorse di calcolo in coda con pesi più elevati, a meno che non si verifichi una delle seguenti condizioni:

- I requisiti di invio di un lavoro corrispondono alla risorsa di elaborazione recuperata.
- Non sono disponibili altre risorse di elaborazione perché sono al massimo della loro capacità.
- `slurmctld` viene riavviato.
- La flotta di AWS ParallelCluster elaborazione viene interrotta e avviata a spegnere e accendere tutti i nodi.

Log correlati

I log relativi agli errori di capacità insufficiente e alla modalità di failover rapido e insufficiente sono disponibili nel Slurm `resume clustermgtd` registro e nel nodo principale.

Slurm **resume** (/var/log/parallelcluster/slurm_resume.log)

Messaggi di errore quando un nodo non viene avviato a causa di una capacità insufficiente.

```
[slurm_plugin.instance_manager:_launch_ec2_instances] - ERROR - Failed RunInstances
request: dcd0c252-90d4-44a7-9c79-ef740f7ecd87
[slurm_plugin.instance_manager:add_instances_for_nodes] - ERROR - Encountered
exception when launching instances for nodes (x1) ['queue1-dy-c-1-1']: An error
occurred
(InsufficientInstanceCapacity) when calling the RunInstances operation (reached max
retries: 1): We currently do not have sufficient p4d.24xlarge capacity in the
Availability Zone you requested (us-west-2b). Our system will be working on
provisioning additional capacity. You can currently get p4d.24xlarge capacity by
not
specifying an Availability Zone in your request or choosing us-west-2a, us-west-2c.
```

Slurm **clustermgtd** (/var/log/parallelcluster/clustermgtd)

La risorsa di calcolo c-1 nella coda 1 è disabilitata a causa della capacità insufficiente.

```
[slurm_plugin.clustermgtd:_reset_timeout_expired_compute_resources] - INFO - The
following compute resources are in down state
due to insufficient capacity: {'queue1': {'c-1':
ComputeResourceFailureEvent(timestamp=datetime.datetime(2022, 4, 14, 23, 0, 4,
769380, tzinfo=datetime.timezone.utc),
error_code='InsufficientInstanceCapacity')}}}, compute resources are reset after
insufficient capacity timeout (600 seconds) expired
```

Dopo la scadenza del timeout di capacità insufficiente, la risorsa di elaborazione viene reimpostata e i nodi all'interno delle risorse di calcolo vengono impostati su. `idle~`

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the
following compute resources because insufficient capacity
timeout expired: {'queue1': ['c-1']}
```

Slurm pianificazione basata sulla memoria

A partire dalla versione 3.2.0, supporta AWS ParallelCluster Slurm pianificazione basata sulla memoria con il parametro di configurazione [SlurmSettings/EnableMemoryBasedScheduling](#)cluster.

 Note

A partire dalla AWS ParallelCluster versione 3.7.0, `EnableMemoryBasedScheduling` può essere abilitata se si configurano più tipi di istanza in `Istanze`.

Per le AWS ParallelCluster versioni da 3.2.0 a 3.6. x, `non` `EnableMemoryBasedScheduling` può essere abilitato se configuri più tipi di istanza in `Istanze`.

 Warning

Quando specifichi più tipi di istanze in un Slurm queue computing resource with `EnableMemoryBasedScheduling` enabled, il `RealMemory` valore è la quantità minima di memoria disponibile per tutti i tipi di istanza. Ciò potrebbe comportare una notevole quantità di memoria inutilizzata se si specificano tipi di istanze con capacità di memoria molto diverse.

Con, `EnableMemoryBasedScheduling: true` Slurm lo scheduler tiene traccia della quantità di memoria richiesta da ciascun processo su ciascun nodo. Quindi, il Slurm lo scheduler utilizza queste informazioni per pianificare più lavori sullo stesso nodo di elaborazione. La quantità totale di memoria richiesta dai job su un nodo non può essere maggiore della memoria disponibile del nodo. Lo scheduler impedisce a un processo di utilizzare più memoria di quella richiesta al momento dell'invio del lavoro.

Con `EnableMemoryBasedScheduling: false`, i job potrebbero competere per la memoria su un nodo condiviso e causare errori ed out-of-memory eventi.

 Warning

Slurm utilizza una notazione di potenza 2 per le sue etichette, ad esempio MB o GB. Leggi queste etichette rispettivamente come MiB e GiB.

Slurm configurazione e pianificazione basata sulla memoria

Con, `EnableMemoryBasedScheduling: true` Slurm imposta quanto segue Slurm parametri di configurazione:

- [SelectTypeParameters=CR_CPU_Memory](#) nella `slurm.conf`. Questa opzione configura la memoria del nodo come risorsa consumabile in Slurm.
- [ConstrainRAMSpace=yes](#) nel `Slurm cgroup.conf`. Con questa opzione, l'accesso di un lavoro alla memoria è limitato alla quantità di memoria richiesta dal lavoro al momento dell'invio.

Note

Diverse altre Slurm i parametri di configurazione possono influire sul comportamento di Slurm scheduler e resource manager quando sono impostate queste due opzioni. Per ulteriori informazioni, consultare la [.Slurm Documentazione](#).

Slurm scheduler e pianificazione basata sulla memoria

EnableMemoryBasedScheduling: false(impostazione predefinita)

Per impostazione predefinita, `EnableMemoryBasedScheduling` è impostato su `false`. Quando è falso, Slurm non include la memoria come risorsa nel suo algoritmo di pianificazione e non tiene traccia della memoria utilizzata dai job. Gli utenti possono specificare l' `--mem MEM_PER_NODE` opzione per impostare la quantità minima di memoria per nodo richiesta da un processo. Ciò impone allo scheduler di scegliere nodi con un `RealMemory` valore almeno pari a 1 `MEM_PER_NODE` durante la pianificazione del lavoro.

Ad esempio, supponiamo che un utente invii due lavori con. `--mem=5GB` Se le risorse richieste come CPUs o GPUs sono disponibili, i processi possono essere eseguiti contemporaneamente su un nodo con 8 GiB di memoria. I due processi non sono pianificati su nodi di calcolo con meno di 5 `RealMemory` GiB di.

Warning

Quando la pianificazione basata sulla memoria è disabilitata, Slurm non tiene traccia della quantità di memoria utilizzata dai job. I lavori eseguiti sullo stesso nodo potrebbero competere per le risorse di memoria e causare il fallimento dell'altro processo.

Quando la pianificazione basata sulla memoria è disabilitata, consigliamo agli utenti di non specificare le `--mem-per-cpu` opzioni o `--mem-per-gpu`. Queste opzioni potrebbero causare un comportamento diverso da quello descritto nella [Slurm Documentazione](#).

EnableMemoryBasedScheduling: true

Quando `EnableMemoryBasedScheduling` è impostato su `true`, Slurm tiene traccia dell'utilizzo della memoria di ogni processo e impedisce ai lavori di utilizzare più memoria di quella richiesta con le opzioni di `--mem` invio.

Utilizzando l'esempio precedente, un utente invia due lavori con. `--mem=5GB` I processi non possono essere eseguiti contemporaneamente su un nodo con 8 GiB di memoria. Questo perché la quantità totale di memoria richiesta è maggiore della memoria disponibile sul nodo.

Con la pianificazione basata sulla memoria abilitata `--mem-per-cpu` e `--mem-per-gpu` comportati in modo coerente con quanto descritto nella Slurm documentazione. Ad esempio, un lavoro viene inviato con. `--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB` In questo caso, Slurm assegna al job un totale di 4 GiB per ogni nodo.

Warning

Quando la pianificazione basata sulla memoria è abilitata, consigliamo agli utenti di includere una `--mem` specifica quando inviano un lavoro. Con l'impostazione predefinita Slurm configurazione inclusa in AWS ParallelCluster, se non è inclusa alcuna opzione di memoria (`--mem`, `--mem-per-cpu`, o `--mem-per-gpu`), Slurm assegna l'intera memoria dei nodi allocati al job, anche se richiede solo una parte delle altre risorse, ad CPUs esempio o. GPUs Ciò impedisce efficacemente la condivisione dei nodi fino al termine del processo, poiché non è disponibile memoria per altri lavori. Ciò accade perché Slurm imposta la memoria per nodo per il lavoro su [DefMemPerNode](#) quando non vengono fornite specifiche di memoria al momento dell'invio del lavoro. Il valore predefinito per questo parametro è 0 e specifica l'accesso illimitato alla memoria di un nodo.

Se nella stessa coda sono disponibili più tipi di risorse di calcolo con diverse quantità di memoria, a un lavoro inviato senza opzioni di memoria potrebbero essere assegnate quantità di memoria diverse su nodi diversi. Dipende dai nodi che lo scheduler mette a disposizione del job. Gli utenti possono definire un valore personalizzato per le opzioni, ad esempio `DefMemPerNode` o [DefMemPerCPU](#), a livello di cluster o partizione in Slurm file di configurazione per prevenire questo comportamento.

Slurm RealMemory e AWS ParallelCluster SchedulableMemory

Con il plugin Slurm configurazione fornita con AWS ParallelCluster, Slurm interpreta [RealMemory](#) come la quantità di memoria per nodo disponibile per i lavori. A partire dalla versione

3.2.0, per impostazione predefinita, è AWS ParallelCluster `RealMemory` impostato sul 95 per cento della memoria elencata in [Amazon EC2 Instance Types](#) e restituita dall' EC2 API [DescribeInstanceTypes](#) Amazon.

Quando la pianificazione basata sulla memoria è disabilitata, Slurm lo scheduler utilizza `RealMemory` per filtrare i nodi quando gli utenti inviano un lavoro con specifiche. `--mem`

Quando la pianificazione basata sulla memoria è abilitata, Slurm lo scheduler interpreta `RealMemory` come la quantità massima di memoria disponibile per i lavori in esecuzione sul nodo di elaborazione.

L'impostazione predefinita potrebbe non essere ottimale per tutti i tipi di istanze:

- Questa impostazione potrebbe essere superiore alla quantità di memoria a cui i nodi possono effettivamente accedere. Ciò può accadere quando i nodi di calcolo sono tipi di istanze di piccole dimensioni.
- Questa impostazione potrebbe essere inferiore alla quantità di memoria a cui i nodi possono effettivamente accedere. Ciò può verificarsi quando i nodi di calcolo sono tipi di istanze di grandi dimensioni e può portare a una notevole quantità di memoria inutilizzata.

È possibile utilizzare [SlurmQueues/ComputeResources/SchedulableMemory](#) per ottimizzare il valore di `configure by` AWS ParallelCluster per i nodi di `RealMemory` calcolo. Per sostituire il valore predefinito, definisci un valore personalizzato `SchedulableMemory` specifico per la configurazione del cluster.

Per verificare la memoria effettiva disponibile di un nodo di calcolo, esegui il `/opt/slurm/sbin/slurmd -C` comando sul nodo. Questo comando restituisce la configurazione hardware del nodo, incluso il [RealMemory](#) valore. Per ulteriori informazioni, consulta [slurmd -C](#).

Assicurati che i processi del sistema operativo del nodo di calcolo dispongano di memoria sufficiente. A tale scopo, limita la memoria disponibile per i job impostando il `SchedulableMemory` valore su un `RealMemory` valore inferiore a quello restituito dal `slurmd -C` comando.

Allocazione di più tipi di istanza con Slurm

A partire dalla AWS ParallelCluster versione 3.3.0, puoi configurare il cluster per l'allocazione dal set di tipi di istanza definiti di una risorsa di calcolo. L'allocazione può essere basata su strategie a basso costo o sulla capacità ottimale della EC2 flotta Amazon.

Questo set di tipi di istanze definiti deve avere tutti lo stesso numero di v CPUs o, se il multithreading è disabilitato, lo stesso numero di core. Inoltre, questo set di tipi di istanze deve

avere lo stesso numero di acceleratori degli stessi produttori. Se [Efa/Enabled](#) è impostato su `true`, le istanze devono avere il supporto EFA. Per ulteriori informazioni e requisiti, vedere [Scheduling//AllocationStrategy](#) e [SlurmQueuesComputeResources/](#). [Instances](#)

Puoi [AllocationStrategy](#) impostarlo su `lowest-price` o in `capacity-optimized` base alla tua [CapacityType](#) configurazione.

In [Instances](#), puoi configurare un set di tipi di istanze.

Note

[A partire dalla AWS ParallelCluster versione 3.7.0, `EnableMemoryBasedScheduling` può essere abilitato se configuri più tipi di istanza in `Istanze`.](#)

Per le AWS ParallelCluster versioni da 3.2.0 a 3.6. [x](#), [non](#)

[EnableMemoryBasedScheduling](#) può essere abilitato se configuri più tipi di istanza in `Istanze`.

Gli esempi seguenti mostrano come è possibile interrogare i tipi di istanza per vCPUs, il supporto EFA e l'architettura.

Query `InstanceTypes` con architettura 96 v CPUs e `x86_64`.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-vcpus,Values=96" "Name=processor-info.supported-
architecture,Values=x86_64" \
  --query "sort_by(InstanceTypes[*].
{InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
&InstanceType})" \
  --output table
```

Query `InstanceTypes` con 64 core, supporto EFA e architettura `arm64`.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-cores,Values=64" "Name=processor-
info.supported-architecture,Values=arm64" "Name=network-info.efa-
supported,Values=true" --query "sort_by(InstanceTypes[*].
{InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
&InstanceType})" \
  --output table
```

Il prossimo esempio di frammento di configurazione del cluster mostra come utilizzarli InstanceType e AllocationStrategy proprietà.

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue-1
      CapacityType: ONDEMAND
      AllocationStrategy: lowest-price
      ...
      ComputeResources:
        - Name: computeresource1
          Instances:
            - InstanceType: r6g.2xlarge
            - InstanceType: m6g.2xlarge
            - InstanceType: c6g.2xlarge
          MinCount: 0
          MaxCount: 500
        - Name: computeresource2
          Instances:
            - InstanceType: m6g.12xlarge
            - InstanceType: x2gd.12xlarge
          MinCount: 0
          MaxCount: 500
      ...
```

Scalabilità del cluster per nodi dinamici

ParallelCluster supporta Slurm i metodi per scalare dinamicamente i cluster utilizzando il plug-in power Slurm saver. Per ulteriori informazioni, consulta la [Cloud Scheduling Guide](#) e la [SlurmPower Saving Guide](#) nella documentazione. Slurm I seguenti argomenti descrivono le Slurm strategie per ogni versione.

Argomenti

- [Slurmstrategie di allocazione dinamica dei nodi nella versione 3.8.0](#)
- [Slurmstrategie di allocazione dinamica dei nodi nella versione 3.7.x](#)
- [Slurmstrategie di allocazione dinamica dei nodi nella versione 3.6.x e precedenti](#)

Slurmstrategie di allocazione dinamica dei nodi nella versione 3.8.0

A partire dalla ParallelCluster versione 3.8.0, ParallelCluster utilizza la scalabilità a livello di processo o la scalabilità a livello di processo come strategia di allocazione dinamica dei nodi predefinita per ParallelCluster scalare il cluster: ridimensiona il cluster in base ai requisiti di ciascun job, al numero di nodi allocati al job e ai nodi che devono essere ripristinati. ParallelCluster ottiene queste informazioni dalla variabile di ambiente `SLURM_RESUME_FILE`.

La scalabilità per i nodi dinamici è un processo in due fasi, che prevede il lancio delle istanze EC2 e l'assegnazione delle istanze Amazon EC2 avviate ai nodi. Slurm Ciascuno di questi due passaggi può essere eseguito utilizzando una logica o best-effort. all-or-nothing

Per il lancio delle istanze Amazon EC2:

- all-or-nothing chiama l'API Amazon EC2 di lancio con un target minimo pari alla capacità target totale
- best-effort chiama il lancio dell'API Amazon EC2 con un target minimo pari a 1 e la capacità target totale è uguale alla capacità richiesta

Per l'assegnazione delle istanze Slurm Amazon EC2 ai nodi:

- all-or-nothing assegna istanze Amazon EC2 Slurm ai nodi solo se è possibile assegnare un'istanza Amazon EC2 a ogni nodo richiesto
- best-effort assegna le istanze di Amazon EC2 Slurm ai nodi anche se tutti i nodi richiesti non sono coperti dalla capacità delle istanze di Amazon EC2

Le possibili combinazioni delle strategie di cui sopra si traducono nelle strategie di lancio. ParallelCluster

Example

<caption>The available ParallelCluster strategie di lancio that can be set into the [ScalingStrategy](#) cluster configuration to be used with scalabilità a livello di lavoro are:</caption>

all-or-nothingridimensionamento:

Questa strategia prevede l' AWS ParallelCluster avvio di una chiamata API di avvio di Amazon EC2 per ogni job, che richiede che tutte le istanze necessarie per il corretto avvio dei nodi di calcolo

richiesti. Ciò garantisce la scalabilità del cluster solo quando è disponibile la capacità richiesta per processo, evitando che le istanze inattive rimangano inattive alla fine del processo di scalabilità.

La strategia utilizza una all-or-nothing logica per il lancio delle istanze Amazon EC2 per ogni processo e una all-or-nothing logica per l'assegnazione delle istanze Amazon EC2 ai nodi. Slurm

La strategia raggruppa le richieste di lancio in batch, uno per ogni risorsa di calcolo richiesta e fino a 500 nodi ciascuno. Per le richieste che riguardano più risorse di elaborazione o superano i 500 nodi, elabora in sequenza più batch. ParallelCluster

L'errore del batch di una singola risorsa comporta la chiusura di tutta la capacità inutilizzata associata, garantendo che nessuna istanza rimanga inattiva al termine del processo di scalabilità.

Limitazioni

- Il tempo impiegato per il ridimensionamento è direttamente proporzionale al numero di lavori inviati per esecuzione del programma di curriculum. Slurm
- L'operazione di scalabilità è limitata dal limite dell'account di RunInstances risorse, impostato a 1000 istanze per impostazione predefinita. Questa limitazione è conforme alle politiche AWS di limitazione delle API EC2, per maggiori dettagli, consulta la documentazione sulla limitazione delle API di [Amazon](#) EC2
- Quando invii un lavoro in una risorsa di elaborazione con un singolo tipo di istanza, in una coda che si estende su più zone di disponibilità, la chiamata all'API di avvio all-or-nothing EC2 ha successo solo se tutta la capacità può essere fornita in un'unica zona di disponibilità.
- Quando invii un lavoro in una risorsa di calcolo con più tipi di istanze, in una coda con un'unica zona di disponibilità, la all-or-nothing chiamata all'API di avvio di Amazon EC2 ha successo solo se tutta la capacità può essere fornita da un singolo tipo di istanza.
- Quando invii un lavoro in una risorsa di calcolo con più tipi di istanze, in una coda che si estende su più zone di disponibilità, la chiamata API di lancio di Amazon EC2 all-or-nothing non è supportata ed esegue invece la scalabilità ottimale. ParallelCluster

greedy-all-or-nothing ridimensionamento:

Questa variante della all-or-nothing strategia garantisce comunque la scalabilità del cluster solo quando è disponibile la capacità richiesta per processo, evitando le istanze inattive alla fine del processo di scalabilità, ma implica l'avvio di ParallelCluster una chiamata API di avvio di Amazon EC2 che mira a una capacità target minima di 1, nel tentativo di massimizzare il numero di nodi avviati fino alla capacità richiesta. La strategia utilizza una logica best-effort per il lancio delle istanze

EC2 per tutti i job, oltre alla all-or-nothing logica per l'assegnazione delle istanze Amazon EC2 ai nodi per ogni job. Slurm

La strategia raggruppa le richieste di lancio in batch, uno per ogni risorsa di calcolo richiesta e fino a 500 nodi ciascuno. Per le richieste che riguardano più risorse di elaborazione o superano i 500 nodi, elabora in sequenza più batch. ParallelCluster

Garantisce che nessuna istanza venga lasciata inattiva alla fine del processo di scalabilità, massimizzando la velocità effettiva a scapito di un temporaneo sovradimensionamento durante il processo di scalabilità.

Limitazioni

- È possibile un sovradimensionamento temporaneo, che comporta costi aggiuntivi per le istanze che passano allo stato di esecuzione prima del completamento della scalabilità.
- Si applica lo stesso limite di istanze previsto dalla all-or-nothing strategia, soggetto al limite dell'account di AWS risorse. RunInstances

scalabilità con il massimo sforzo:

Questa strategia prevede una chiamata all'API di avvio di Amazon EC2 mirando a una capacità minima di 1 e mirando a raggiungere la capacità totale richiesta al costo di lasciare le istanze inattive dopo l'esecuzione del processo di scalabilità se non tutta la capacità richiesta è disponibile. La strategia utilizza una logica best-effort per il lancio delle istanze Amazon EC2 per tutti i lavori, oltre alla logica best-effort per l'assegnazione delle istanze Amazon EC2 ai nodi Slurm per ogni processo.

La strategia raggruppa le richieste di avvio in batch, uno per ogni risorsa di calcolo richiesta e fino a 500 nodi ciascuno. Per le richieste che riguardano più risorse di elaborazione o superano i 500 nodi, elabora in sequenza più batch. ParallelCluster

Questa strategia consente di scalare ben oltre il limite predefinito di 1000 istanze su più esecuzioni di processi di scalabilità, al costo di avere istanze inattive tra i diversi processi di scalabilità.

Limitazioni

- Possibili istanze inattive in esecuzione alla fine del processo di scalabilità, nel caso in cui non sia possibile allocare tutti i nodi richiesti dai job.

Di seguito è riportato un esempio che mostra come si comporta il ridimensionamento dei nodi dinamici utilizzando le diverse strategie di lancio. ParallelCluster Supponiamo di aver inviato due

lavori per la richiesta di 20 nodi ciascuno, per un totale di 40 nodi dello stesso tipo, ma che siano disponibili solo 30 istanze Amazon EC2 per coprire la capacità richiesta su EC2.

all-or-nothingridimensionamento:

- Per il primo processo, viene chiamata un'API di avvio di all-or-nothingAmazon EC2, che richiede 20 istanze. Una chiamata riuscita dà come risultato il lancio di 20 istanze
- all-or-nothing l'assegnazione delle 20 istanze avviate ai Slurm nodi per il primo processo ha esito positivo
- Viene chiamata un'altra API di avvio di all-or-nothingAmazon EC2, che richiede 20 istanze per il secondo processo. La chiamata non ha esito positivo, poiché è disponibile solo la capacità per altre 10 istanze. Al momento non viene avviata alcuna istanza

greedy-all-or-nothingridimensionamento:

- Viene chiamata un'API di avvio Amazon EC2 che richiede 40 istanze, ovvero la capacità totale richiesta da tutti i processi. Ciò si traduce nel lancio di 30 istanze
- all-or-nothingL'assegnazione di 20 delle istanze avviate ai Slurm nodi per il primo processo ha esito positivo
- Viene tentata un'altra all-or-nothingassegnazione delle istanze avviate rimanenti ai Slurm nodi per il secondo processo, ma poiché ci sono solo 10 istanze disponibili su un totale di 20 richieste dal processo, l'assegnazione non ha esito positivo
- Le 10 istanze avviate non assegnate vengono terminate

scalabilità con il massimo sforzo:

- Viene chiamata un'API di avvio Amazon EC2 che richiede 40 istanze, ovvero la capacità totale richiesta da tutti i processi. Ciò si traduce nel lancio di 30 istanze.
- L'assegnazione ottimale di 20 delle istanze lanciate ai Slurm nodi per il primo processo ha esito positivo.
- Un'altra assegnazione ottimale delle restanti 10 istanze avviate ai Slurm nodi per il secondo processo ha esito positivo, anche se la capacità totale richiesta era di 20. Tuttavia, poiché il processo richiedeva i 20 nodi ed era possibile assegnare istanze Amazon EC2 solo a 10 di essi, il processo non può essere avviato e le istanze vengono lasciate inattive, finché non viene trovata una capacità sufficiente per avviare le 10 istanze mancanti in una chiamata successiva

del processo di scalabilità, oppure lo scheduler pianifica il processo su altri nodi di calcolo già in esecuzione.

Slurmstrategie di allocazione dinamica dei nodi nella versione 3.7.x

ParallelCluster utilizza 2 tipi di strategie di allocazione dinamica dei nodi per scalare il cluster:

- Allocazione basata sulle informazioni richieste disponibili sui nodi:
 - Ripresa di tutti i nodi o ridimensionamento dell'elenco dei nodi:

ParallelCluster ridimensiona il cluster in base solo ai nomi degli elenchi Slurm di nodi richiesti quando viene eseguito. Slurm ResumeProgram Alloca le risorse di calcolo ai nodi solo in base al nome del nodo. L'elenco dei nomi dei nodi può includere più processi.

- Curriculum a livello di lavoro o scalabilità a livello di lavoro:

ParallelCluster ridimensiona il cluster in base ai requisiti di ciascun processo, al numero corrente di nodi allocati al processo e ai nodi che devono essere ripristinati. ParallelCluster ottiene queste informazioni dalla variabile di ambiente. `SLURM_RESUME_FILE`

- Allocazione con una strategia di lancio di Amazon EC2:
 - Scalabilità al massimo:

ParallelCluster ridimensiona il cluster utilizzando una chiamata API di avvio di Amazon EC2 con la capacità di destinazione minima pari a 1, per avviare alcune, ma non necessariamente tutte, le istanze necessarie per supportare i nodi richiesti.

- Una scalabilità: Il-or-nothing

ParallelCluster aumenta la scalabilità del cluster utilizzando una chiamata API di avvio di Amazon EC2 che ha esito positivo solo se vengono lanciate tutte le istanze necessarie per supportare i nodi richiesti. In questo caso, chiama l'API dell'istanza di avvio di Amazon EC2 con la capacità target minima pari alla capacità totale richiesta.

Per impostazione predefinita, ParallelCluster utilizza il ridimensionamento dell'elenco dei nodi con una strategia di lancio di Amazon EC2 al massimo per avviare alcune, ma non necessariamente tutte, le istanze necessarie per supportare i nodi richiesti. Cerca di fornire quanta più capacità possibile per soddisfare il carico di lavoro inviato.

A partire dalla ParallelCluster versione 3.7.0, ParallelCluster utilizza la scalabilità a livello di processo con una strategia di lancio all-or-nothingEC2 per i lavori inviati in modalità esclusiva. Quando invii un lavoro in modalità esclusiva, il lavoro ha accesso esclusivo ai nodi allocati. Per ulteriori informazioni, consulta [EXCLUSIVE](#) nella Slurm documentazione.

Per inviare un'offerta di lavoro in modalità esclusiva:

- Passa il flag esclusivo quando invii un Slurm lavoro al cluster. Ad esempio, `sbatch ... --exclusive`.
- O
- Invia un lavoro a una coda del cluster che è stata configurata con [JobExclusiveAllocation](#) set to `true`

Quando invii un lavoro in modalità esclusiva:

- ParallelCluster attualmente i batch lanciano richieste per includere fino a 500 nodi. Se un processo richiede più di 500 nodi, ParallelCluster effettua una richiesta di all-or-nothingavvio per ogni set di 500 nodi e una richiesta di avvio aggiuntiva per il resto dei nodi.
- Se l'allocazione dei nodi è in una singola risorsa di elaborazione, ParallelCluster effettua una richiesta di all-or-nothingavvio per ogni set di 500 nodi e una richiesta di avvio aggiuntiva per il resto dei nodi. Se una richiesta di avvio fallisce, ParallelCluster termina la capacità inutilizzata creata da tutte le richieste di avvio.
- Se l'allocazione dei nodi si estende su più risorse di elaborazione, ParallelCluster deve effettuare una richiesta di all-or-nothingavvio per ogni risorsa di calcolo. Anche queste richieste vengono raggruppate. Se una richiesta di avvio fallisce per una delle risorse di calcolo, ParallelCluster termina la capacità inutilizzata creata da tutte le richieste di avvio delle risorse di calcolo.

scalabilità a livello di lavoro con limitazioni note della strategia di lancio: all-or-nothing

- Quando invii un lavoro in una risorsa di calcolo con un singolo tipo di istanza, in una coda che si estende su più zone di disponibilità, la chiamata all'API di avvio di all-or-nothingEC2 ha esito positivo solo se tutta la capacità può essere fornita in un'unica zona di disponibilità.
- Quando invii un lavoro in una risorsa di calcolo con più tipi di istanze, in una coda con un'unica zona di disponibilità, la all-or-nothingchiamata all'API di avvio di Amazon EC2 ha successo solo se tutta la capacità può essere fornita da un singolo tipo di istanza.

- Quando invii un lavoro in una risorsa di calcolo con più tipi di istanze, in una coda che si estende su più zone di disponibilità, la chiamata API di lancio di Amazon EC2 all-or-nothing non è supportata ed esegue invece la scalabilità ottimale. ParallelCluster

Slurmstrategie di allocazione dinamica dei nodi nella versione 3.6.x e precedenti

AWS ParallelCluster utilizza solo un tipo di strategia di allocazione dinamica dei nodi per scalare il cluster:

- Allocazione basata sulle informazioni richieste disponibili sul nodo:
 - Ripresa di tutti i nodi o ridimensionamento dell'elenco dei nodi: ParallelCluster ridimensiona il cluster in base solo ai nomi degli elenchi Slurm di nodi richiesti quando viene eseguito. Slurm ResumeProgram Alloca le risorse di calcolo ai nodi solo in base al nome del nodo. L'elenco dei nomi dei nodi può includere più processi.
- Allocazione con una strategia di lancio di Amazon EC2:
 - Scalabilità ottimale: aumenta la ParallelCluster scalabilità del cluster utilizzando una chiamata API di avvio di Amazon EC2 con la capacità target minima pari a 1, per avviare alcune, ma non necessariamente tutte, le istanze necessarie per supportare i nodi richiesti.

ParallelCluster utilizza il ridimensionamento dell'elenco dei nodi con una strategia di lancio di Amazon EC2 al massimo per avviare alcune, ma non necessariamente tutte, le istanze necessarie per supportare i nodi richiesti. Cerca di fornire quanta più capacità possibile per soddisfare il carico di lavoro inviato.

Limitazioni

- Possibili istanze inattive in esecuzione al termine del processo di scalabilità, nel caso in cui non sia possibile allocare tutti i nodi richiesti dai job.

Slurmcontabilità con AWS ParallelCluster

A partire dalla versione 3.3.0, AWS ParallelCluster supporta la Slurm contabilità con il parametro di configurazione del cluster [SlurmSettings/Database](#).

A partire dalla versione 3.10.0, AWS ParallelCluster supporta la Slurm contabilità con uno Slurmdbd esterno con il parametro di configurazione del cluster/. [SlurmSettingsExternalSlurmdbd](#) L'utilizzo di uno Slurmdbd esterno è consigliato se più cluster condividono lo stesso database.

Con la Slurm contabilità, puoi integrare un database contabile esterno per eseguire le seguenti operazioni:

- Gestisci utenti o gruppi di utenti del cluster e altre entità. Con questa funzionalità, puoi utilizzare le funzionalità più avanzate Slurm di cui dispone, ad esempio, l'applicazione dei limiti delle risorse, la condivisione equa e. QOSs
- Raccogli e salva i dati del lavoro, come l'utente che ha eseguito il lavoro, la durata del lavoro e le risorse utilizzate. È possibile visualizzare i dati salvati con l'`sacct` utilità.

 Note

AWS ParallelCluster supporta la Slurm contabilità per i [Slurmserver di database MySQL supportati](#).

Utilizzo della Slurm contabilità tramite external Slurmdbd nella versione 3.10.0 e AWS ParallelCluster successive

Prima di configurare la Slurm contabilità, è necessario disporre di un server di Slurmdbd database esterno esistente, che si connette a un server di database esterno esistente.

Per configurarlo, definisci quanto segue:

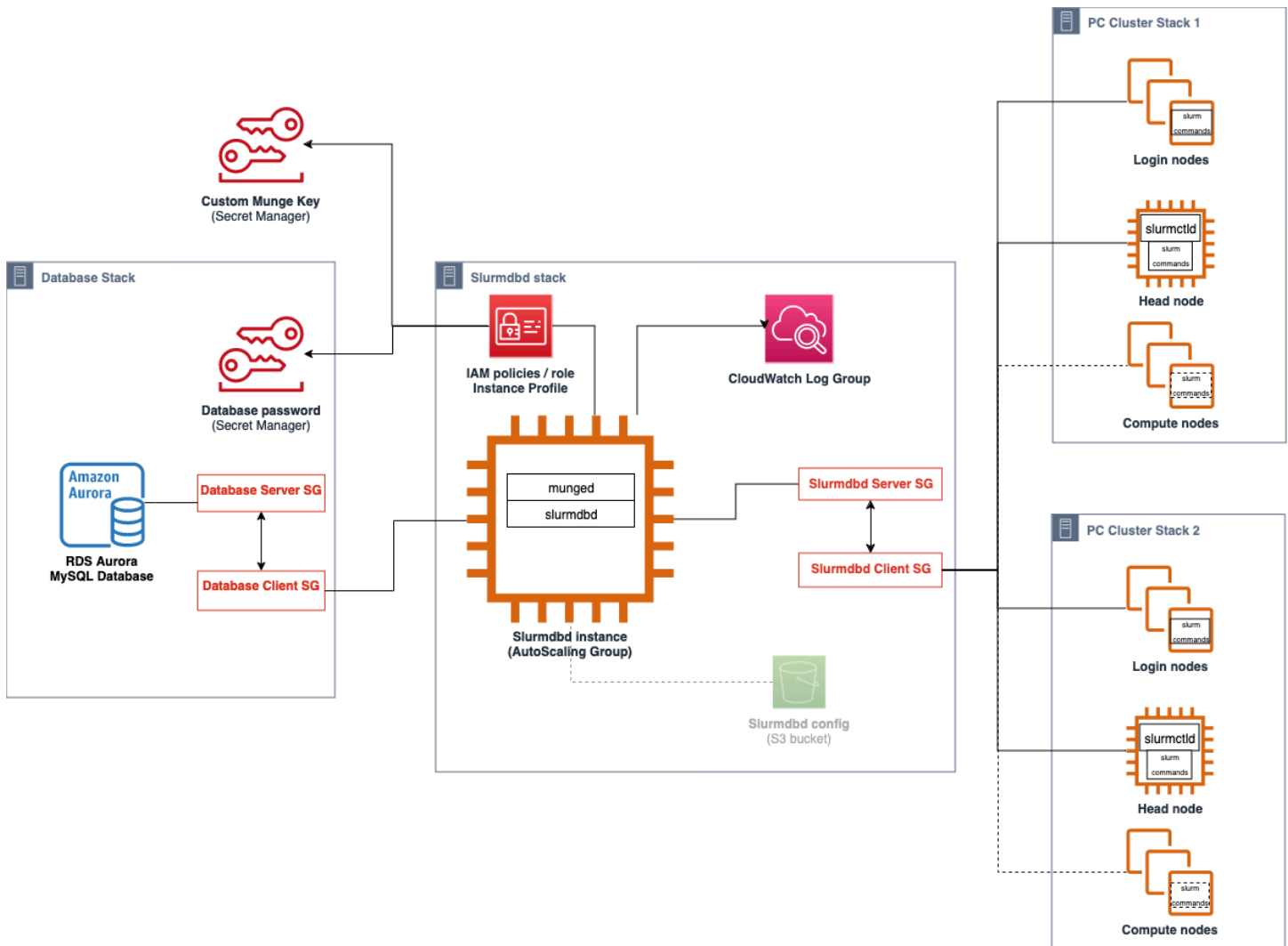
- L'indirizzo del Slurmdbd server esterno in [ExternalSlurmdbd/Host](#). Il server deve esistere ed essere raggiungibile dal nodo principale.
- La chiave munge per comunicare con il Slurmdbd server esterno. [MungeKeySecretArn](#)

Per seguire un tutorial, consulta [Creazione di un cluster con un cluster esterno Slurmdbd contabilità](#).

 Note

L'utente è responsabile della gestione delle entità contabili del Slurm database.

L'architettura della funzionalità di SlurmDB supporto AWS ParallelCluster esterno consente a più cluster di condividere SlurmDB lo stesso database.



⚠ Warning

Il traffico tra AWS ParallelCluster e l'esterno non SlurmDB è crittografato. Si consiglia di eseguire il cluster e quello esterno SlurmDB in una rete affidabile.

Utilizzo della Slurm contabilità utilizzando head node Slurmdbd nella AWS ParallelCluster versione 3.3.0 e successive

Prima di configurare la Slurm contabilità, è necessario disporre di un server di database esterno e di un database esistenti che mysql utilizzino il protocollo.

Per configurare la Slurm contabilità con AWS ParallelCluster, è necessario definire quanto segue:

- L'URI per il server di database esterno in [Database/Uri](#). Il server deve esistere ed essere raggiungibile dal nodo principale.
- Credenziali per accedere al database esterno definite in [Database/PasswordSecretArne Database/](#). [UserName](#) AWS ParallelCluster utilizza queste informazioni per configurare la contabilità a Slurm livello e il `slurmdbd` servizio sul nodo principale. `slurmdbd` è il demone che gestisce la comunicazione tra il cluster e il server del database.

Per seguire un tutorial, vedere. [Creazione di un cluster con Slurm contabilità](#)

Note

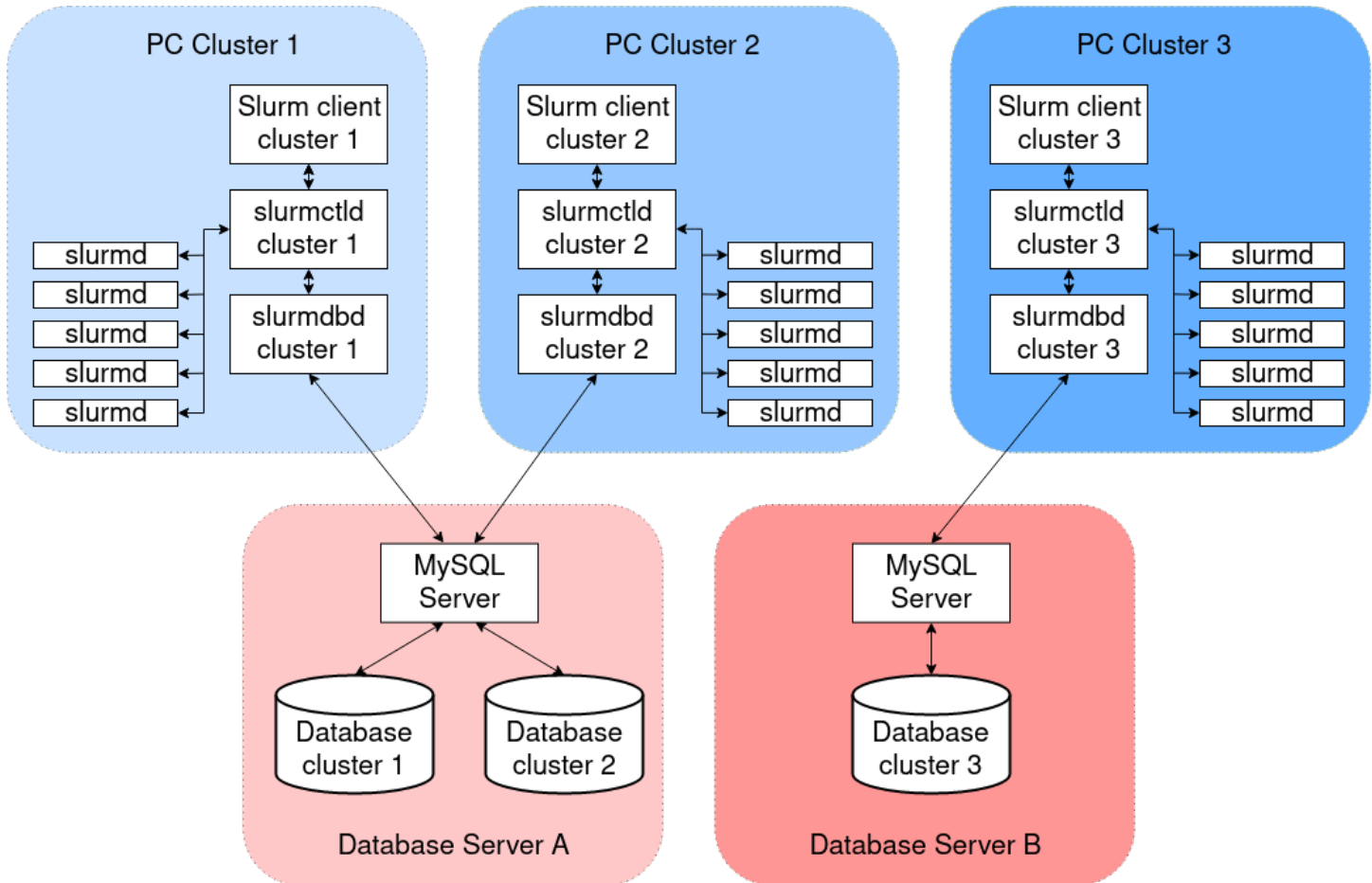
AWS ParallelCluster esegue un bootstrap di base del database di Slurm contabilità impostando l'utente predefinito del cluster come amministratore del database nel Slurm database. AWS ParallelCluster non aggiunge nessun altro utente al database di contabilità. Il cliente è responsabile della gestione delle entità contabili nel Slurm database.

AWS ParallelCluster configura `slurmdbd` per garantire che un cluster disponga di un proprio Slurm database sul server del database. Lo stesso server di database può essere utilizzato su più cluster, ma ogni cluster ha il proprio database separato. AWS ParallelCluster utilizza il nome del cluster per definire il nome del database nel `StorageLoc` parametro del file `slurmdbd` di configurazione. Considerate la situazione seguente. Un database presente sul server di database include un nome di cluster che non corrisponde a un nome di cluster attivo. In questo caso, puoi creare un nuovo cluster con quel nome di cluster da mappare a quel database. Slurm riutilizza il database per il nuovo cluster.

Warning

- Non è consigliabile configurare più di un cluster per utilizzare lo stesso database contemporaneamente. Ciò può causare problemi di prestazioni o persino situazioni di deadlock del database.
- Se l'`Slurmaccounting` è abilitato sul nodo principale di un cluster, consigliamo di utilizzare un tipo di istanza con una CPU potente, più memoria e una maggiore larghezza di banda di rete. Slurm la contabilità può mettere a dura prova il nodo principale del cluster.

Nell'attuale architettura della funzionalità di AWS ParallelCluster Slurm contabilità, ogni cluster ha la propria istanza del `slurmdbd` demone, come illustrato nel seguente diagramma di configurazione.



Se si aggiungono funzionalità Slurm multi-cluster o federative personalizzate all'ambiente cluster, tutti i cluster devono fare riferimento alla stessa istanza. `slurmdbd` In alternativa, ti consigliamo di abilitare l' AWS ParallelCluster Slurmaccounting su un cluster e di configurare manualmente gli altri cluster per la connessione a `slurmdbd` quelli ospitati sul primo cluster.

Se utilizzi AWS ParallelCluster versioni precedenti alla versione 3.3.0, fai riferimento al metodo alternativo per implementare la Slurm contabilità descritto in questo post del blog [HPC](#).

Slurmconsiderazioni contabili

Database e cluster su diversi VPCs

Per abilitare la Slurm contabilità, è necessario un server di database che funga da backend per le operazioni di lettura e scrittura eseguite dal `slurmdbd` demone. Prima che il cluster venga creato o

aggiornato per abilitare la Slurm contabilità, il nodo principale deve essere in grado di raggiungere il server del database.

Se devi implementare il server di database su un VPC diverso da quello utilizzato dal cluster, considera quanto segue:

- Per abilitare la comunicazione tra il `slurmd` lato del cluster e il server del database, è necessario configurare la connettività tra i due VPCs. Per ulteriori informazioni, consulta [VPC Peering](#) nella Amazon Virtual Private Cloud User Guide.
- È necessario creare il gruppo di sicurezza che si desidera collegare al nodo principale sul VPC del cluster. Dopo il peering dei due VPCs gruppi, è disponibile il collegamento incrociato tra i gruppi di sicurezza lato database e lato cluster. Per ulteriori informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Configurazione della crittografia TLS tra `slurmd` e il server del database

Con la configurazione di Slurm contabilità predefinita che AWS ParallelCluster fornisce, `slurmd` stabilisce una connessione crittografata TLS al server del database, se il server supporta la crittografia TLS. AWS servizi di database come Amazon RDS e Amazon Aurora supportano la crittografia TLS per impostazione predefinita.

È possibile richiedere connessioni sicure sul lato server impostando il `require_secure_transport` parametro sul server del database. Questo è configurato nel CloudFormation modello fornito.

Seguendo le migliori pratiche di sicurezza, si consiglia di abilitare anche la verifica dell'identità del server sul `slurmd` client. A tale scopo, configura `StorageParameters` in `slurmd.conf`. Carica il certificato CA del server nel nodo principale del cluster. Quindi, imposta l'opzione `SSL_CA` di `StorageParameters` in `slurmd.conf` sul percorso del certificato CA del server sul nodo principale. In questo modo si abilita la verifica dell'identità del server sul lato. `slurmd` Dopo aver apportato queste modifiche, riavvia il `slurmd` servizio per ristabilire la connettività al server del database con la verifica dell'identità abilitata.

Aggiornamento delle credenziali del database

Per aggiornare i valori di `Database/Username` o `PasswordSecretArn`, devi prima interrompere il parco di elaborazione. Supponiamo che il valore segreto memorizzato nel Gestione dei segreti AWS segreto venga modificato e il relativo ARN non venga modificato. In questa situazione, il cluster non

aggiorna automaticamente la password del database con il nuovo valore. Per aggiornare il cluster per il nuovo valore segreto, esegui il comando seguente dal nodo principale.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

Warning

Per evitare di perdere i dati contabili, si consiglia di modificare la password del database solo quando la flotta di elaborazione viene interrotta.

Monitoraggio del database

Si consiglia di abilitare le funzionalità di monitoraggio dei servizi di AWS database. Per ulteriori informazioni, consulta la [documentazione sul monitoraggio di Amazon RDS](#) o [Amazon Aurora](#).

Slurm personalizzazione della configurazione

A partire dalla AWS ParallelCluster versione 3.6.0, è possibile personalizzare `slurm.conf` Slurm configurazione in una configurazione di AWS ParallelCluster cluster.

Nella configurazione del cluster, è possibile personalizzare Slurm parametri di configurazione utilizzando le seguenti impostazioni di configurazione del cluster:

- Personalizzazione Slurm parametri per l'intero cluster utilizzando il parametro [SlurmSettings/CustomSlurmSettings](#) o il [CustomSlurmSettingsIncludeFile](#) parametro. AWS ParallelCluster fallisce se si specificano entrambi.
- Personalizzazione Slurm parametri per una coda utilizzando [SlurmQueues/CustomSlurmSettings](#) (mappati a Slurm partizioni).
- Personalizzazione Slurm parametri per una risorsa di calcolo utilizzando [SlurmQueuesComputeResources/CustomSlurmSettings](#) (mappati a Slurm nodi).

Slurm limiti e considerazioni sulla personalizzazione della configurazione durante l'utilizzo AWS ParallelCluster

- Per `CustomSlurmSettings` e `CustomSlurmSettingsIncludeFile` impostazioni, è possibile specificare e aggiornare solo `slurm.conf` i parametri inclusi nel [Slurm versione](#) supportata dalla AWS ParallelCluster versione utilizzata per configurare un cluster.

- Se si specifica personalizzato Slurm configurazioni in qualsiasi CustomSlurmSettings parametro, AWS ParallelCluster esegue controlli di convalida e impedisce l'impostazione o l'aggiornamento Slurm parametri di configurazione che sono in conflitto con AWS ParallelCluster la logica. Il Slurm i parametri di configurazione con cui è noto che sono in conflitto AWS ParallelCluster vengono identificati negli elenchi di rifiuto. Le liste di rifiuto possono cambiare nelle AWS ParallelCluster versioni future se altre Slurm vengono aggiunte funzionalità. Per ulteriori informazioni, consulta [Non inserito nell'elenco Slurm parametri di configurazione per CustomSlurmSettings](#).
- AWS ParallelCluster controlla solo se un parametro è in una lista di rifiuto. AWS ParallelCluster non convalida la tua personalizzazione Slurm sintassi o semantica dei parametri di configurazione. Sei responsabile della convalida della tua personalizzazione Slurm parametri di configurazione. Personalizzazione non valida Slurm i parametri di configurazione possono causare Slurm errori dei daemon che possono causare errori di creazione e aggiornamento del cluster.
- Se si specifica personalizzato Slurm configurazioni in CustomSlurmSettingsIncludeFile, AWS ParallelCluster non esegue alcuna convalida.
- Puoi aggiornare CustomSlurmSettings e CustomSlurmSettingsIncludeFile senza interrompere e avviare la flotta di elaborazione. In questo caso, AWS ParallelCluster riavvia il slurmctld demone ed esegue il comando. `scontrol reconfigure`

Medio Slurm i parametri di configurazione potrebbero richiedere diverse operazioni prima che una modifica venga registrata nell'intero cluster. Ad esempio, potrebbero richiedere il riavvio di tutti i demoni del cluster. L'utente ha la responsabilità di verificare se AWS ParallelCluster le operazioni sono sufficienti per diffondere la personalizzazione Slurm impostazioni dei parametri di configurazione durante gli aggiornamenti. Se ritieni che AWS ParallelCluster le operazioni non siano sufficienti, è tua responsabilità fornire le azioni aggiuntive necessarie per propagare le impostazioni aggiornate, come consigliato nella [Slurm documentazione](#).

Non inserito nell'elenco Slurm parametri di configurazione per **CustomSlurmSettings**

Le tabelle seguenti elencano i parametri con le AWS ParallelCluster versioni che ne negano l'uso, a partire dalla versione 3.6.0. CustomSlurmSettings non è supportato per AWS ParallelCluster le versioni precedenti alla versione 3.6.0.

Parametri negati a livello di cluster:

Slurm parametro	Deny-list nelle versioni AWS ParallelCluster
CommunicationParameters	3.6.0
Epilogo	3.6.0
GresTypes	3.6.0
LaunchParameters	3.6.0
Prolog	3.6.0
ReconfigFlags	3.6.0
ResumeFailProgram	3.6.0
ResumeProgram	3.6.0
ResumeTimeout	3.6.0
SlurmctldHost	3.6.0
SlurmctldLogFile	3.6.0
SlurmctldParameters	3.6.0
SlurmdLogfile	3.6.0
SlurmUser	3.6.0
SuspendExcNodes	3.6.0
SuspendProgram	3.6.0
SuspendTime	3.6.0
TaskPlugin	3.6.0
TreeWidth	3.6.0

Parametri negati a livello di cluster quando sono nativi [Slurm l'integrazione contabile](#) è configurata nella configurazione del cluster:

Slurm parametro	Deny-list nelle versioni AWS ParallelCluster
AccountingStorageType	3.6.0
AccountingStorageHost	3.6.0
AccountingStoragePort	3.6.0
AccountingStorageUser	3.6.0
JobAcctGatherType	3.6.0

Parametri negati a livello di coda (partizione) per le code gestite da: AWS ParallelCluster

Slurm parametro	Negate nelle versioni AWS ParallelCluster
Nodi	3.6.0
PartitionName	3.6.0
ResumeTimeout	3.6.0
Stato	3.6.0
SuspendTime	3.6.0

Parametri negati a livello di risorsa di calcolo (nodo) per la risorsa di elaborazione gestita da: AWS ParallelCluster

Slurm parametro	Deny-list nella versione e nelle versioni successive AWS ParallelCluster
CPUs	3.6.0
Funzionalità	3.6.0
Gres	3.6.0

Slurm parametro	Deny-list nella versione e nelle versioni successive AWS ParallelCluster
NodeAddr	3.6.0
NodeHostname	3.6.0
NodeName	3.6.0
Weight	3.7.0

Slurm e **prolog epilog**

A partire dalla AWS ParallelCluster versione 3.6.0, la Slurm configurazione con cui viene distribuita AWS ParallelCluster include i Prolog parametri di configurazione e: Epilog

```
# PROLOG AND EPILOG
Prolog=/opt/slurm/etc/scripts/prolog.d/*
Epilog=/opt/slurm/etc/scripts/epilog.d/*
SchedulerParameters=nohold_on_prolog_fail
BatchStartTimeout=180
```

Per ulteriori informazioni, consulta la Guida [Prolog ed Epilog](#) nella documentazione. Slurm

AWS ParallelCluster include i seguenti script di prolog ed epilog:

- 90_plcluster_health_check_manager(nella cartella) Prolog
- 90_pcluster_noop(nella Epilog cartella)

Note

Prolog Sia la Epilog cartella che la cartella devono contenere almeno un file.

È possibile utilizzare epilog script personalizzati prolog o personalizzati aggiungendoli alle Epilog cartelle Prolog e corrispondenti.

 Warning

Slurmesegue tutti gli script nelle cartelle, in ordine alfabetico inverso.

La durata di esecuzione degli epilog script prolog and influisce sul tempo necessario per eseguire un processo. Aggiorna le impostazioni BatchStartTimeout di configurazione quando esegui prolog script multipli o di lunga durata. L'impostazione predefinita è 3 minuti.

Se utilizzi epilog script prolog e personalizzati, individua gli script nelle Prolog rispettive Epilog cartelle. Si consiglia di conservare lo 90_plcluster_health_check_manager script che viene eseguito prima di ogni script personalizzato. Per ulteriori informazioni, consulta [Slurm personalizzazione della configurazione](#).

Dimensioni e aggiornamento della capacità del cluster

La capacità del cluster è definita dal numero di nodi di calcolo scalabili dal cluster. I nodi di calcolo sono supportati da EC2 istanze Amazon definite all'interno delle risorse di elaborazione nella AWS ParallelCluster configurazione (Scheduling/SlurmQueues/ [ComputeResources](#)) e sono organizzati in code (Scheduling/[SlurmQueues](#)) con mappatura 1:1 a Slurm partizioni.

All'interno di una risorsa di calcolo è possibile configurare il numero minimo di nodi di calcolo (istanze) che devono essere sempre mantenuti in esecuzione nel cluster (MinCount) e il numero massimo di istanze che la risorsa di calcolo può scalare fino a (3). MaxCount

Al momento della creazione del cluster o dopo un aggiornamento del cluster, AWS ParallelCluster avvia tutte le EC2 istanze Amazon configurate MinCount per ogni risorsa di calcolo (Scheduling/SlurmQueues/ [ComputeResources](#)) definita nel cluster. Le istanze avviate per coprire il numero minimo di nodi per le risorse di calcolo nel cluster sono chiamate nodi statici. Una volta avviati, i nodi statici sono pensati per essere persistenti nel cluster e non vengono terminati dal sistema, a meno che non si verifichi un evento o una condizione particolare. Tali eventi includono, ad esempio, il fallimento di Slurm o i controlli EC2 sanitari di Amazon e la modifica del Slurm stato del nodo su DRAIN o DOWN.

Le EC2 istanze Amazon, nell'intervallo da 0 **1** a '**MaxCount - MinCount**' (**MaxCount** meno) **MinCount**), lanciate su richiesta per gestire l'aumento del carico del cluster, vengono chiamate nodi dinamici. La loro natura è effimera, vengono avviate per eseguire lavori in sospeso e vengono terminate quando rimangono inattive per un periodo di tempo definito Scheduling/

SlurmSettings/[ScaledownIdleTime](#) nella configurazione del cluster (impostazione predefinita: 10 minuti).

I nodi statici e i nodi dinamici sono conformi al seguente schema di denominazione:

- Nodi statici dove `<Queue/Name>-st-<ComputeResource/Name>-<num> <num> = 1..ComputeResource/MinCount`
- Nodi dinamici `<Queue/Name>-dy-<ComputeResource/Name>-<num>` dove `<num> = 1..(ComputeResource/MaxCount - ComputeResource/MinCount)`

Ad esempio, data la seguente AWS ParallelCluster configurazione:

```
Scheduling:
  Scheduler: Slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

I seguenti nodi verranno definiti in Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up      infinite    50   idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up      infinite   100   idle queue1-st-c5xlarge-[1-100]
```

Quando una risorsa di elaborazione lo ha `MinCount == MaxCount`, tutti i nodi di calcolo corrispondenti saranno statici e tutte le istanze verranno avviate al momento della creazione/aggiornamento del cluster e mantenute attive e funzionanti. Per esempio:

```
Scheduling:
  Scheduler: slurm
```

```
SlurmQueues:
- Name: queue1
  ComputeResources:
    - Name: c5xlarge
      Instances:
        - InstanceType: c5.xlarge
      MinCount: 100
      MaxCount: 100
```

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    100   idle queue1-st-c5xlarge-[1-100]
```

Aggiornamento della capacità del cluster

L'aggiornamento della capacità del cluster include l'aggiunta o la rimozione di code, risorse di elaborazione o la modifica MinCount/MaxCount di una risorsa di elaborazione. A partire dalla AWS ParallelCluster versione 3.9.0, la riduzione delle dimensioni di una coda richiede che il parco di elaborazione venga interrotto o [QueueUpdateStrategy](#) impostato su TERMINATE prima dell'aggiornamento del cluster. Non è necessario interrompere la flotta di elaborazione o impostare su TERMINATE quando: [QueueUpdateStrategy](#)

- Aggiungere nuove code a Scheduling/ [SlurmQueues](#)
- Aggiungere nuove risorse di calcolo a una coda Scheduling/ [SlurmQueues/ComputeResources](#)
- Aumento della potenza [MaxCount](#) di una risorsa di elaborazione
- Aumento MinCount di una risorsa di elaborazione e aumento MaxCount della stessa risorsa di elaborazione almeno della stessa quantità

Considerazioni e limitazioni

Questa sezione ha lo scopo di delineare eventuali fattori, vincoli o limitazioni importanti da tenere in considerazione durante il ridimensionamento della capacità del cluster.

- Quando si rimuove una coda da Scheduling/[SlurmQueues](#) tutti i nodi di elaborazione con nome<Queue/Name>-*, sia statici che dinamici, verrà rimossa dal Slurm la configurazione e le EC2 istanze Amazon corrispondenti verranno terminate.
- Quando si rimuove una risorsa di calcolo Scheduling/SlurmQueues/[ComputeResources](#) da una coda, tutti i nodi di calcolo con nome<Queue/Name>-*-<ComputeResource/Name>-*, sia statici che dinamici, verranno rimossi dalla Slurm la configurazione e le EC2 istanze Amazon corrispondenti verranno terminate.

Quando si modifica il MinCount parametro di una risorsa di calcolo, possiamo distinguere due diversi scenari, se MaxCount viene mantenuto uguale a MinCount (solo capacità statica) e se MaxCount è maggiore di MinCount (capacità statica e dinamica mista).

La capacità cambia solo con nodi statici

- SeMinCount == MaxCount, aumentando MinCount (eMaxCount), il cluster verrà configurato estendendo il numero di nodi statici al nuovo valore di MinCount <Queue/Name>-st-<ComputeResource/Name>-<new_MinCount> e il sistema continuerà a provare ad avviare EC2 istanze Amazon per soddisfare la nuova capacità statica richiesta.
- SeMinCount == MaxCount, diminuendo MinCount (eMaxCount) la quantità N, il cluster verrà configurato rimuovendo gli ultimi N nodi statici <Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>] e il sistema interromperà le istanze Amazon EC2 corrispondenti.

- Stato iniziale MinCount = MaxCount = 100

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    100   idle queue1-st-c5xlarge-[1-100]
```

- Aggiornamento -30 su MinCount e MaxCount: MinCount = MaxCount = 70

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    70    idle queue1-st-c5xlarge-[1-70]
```

Modifiche di capacità con nodi misti

Se $\text{MinCount} < \text{MaxCount}$, aumentando MinCount di una quantità N (supponendo che MaxCount venga mantenuto invariato), il cluster verrà configurato estendendo il numero di nodi statici al nuovo valore di MinCount ($\text{old\_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N>` e il sistema continuerà a provare ad avviare EC2 istanze Amazon per soddisfare la nuova capacità statica richiesta. Inoltre, per rispettare la MaxCount capacità della risorsa di calcolo, la configurazione del cluster viene aggiornata rimuovendo gli ultimi N nodi dinamici: `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old\_MinCount - N>...<MaxCount - old\_MinCount>]` e il sistema interromperà le istanze Amazon EC2 corrispondenti.

- Stato iniziale: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Aggiorna +30 a MinCount : $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*    up       infinite   130    idle queue1-st-c5xlarge-[1-130]
```

Se $\text{MinCount} < \text{MaxCount}$, aumentando MinCount e MaxCount della stessa quantità N , il cluster verrà configurato estendendo il numero di nodi statici al nuovo valore di MinCount ($\text{old\_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N>` e il sistema continuerà a provare ad avviare EC2 istanze Amazon per soddisfare la nuova capacità statica richiesta. Inoltre, non verrà apportata alcuna modifica al numero di nodi dinamici per rispettare il nuovo

MaxCount value.

- Stato iniziale: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Aggiorna +30 a MinCount : MinCount = 130 (MaxCount = 180)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   130    idle queue1-st-c5xlarge-[1-130]
```

Se $\text{MinCount} < \text{MaxCount}$, diminuendo la MinCount quantità N (supponendo che MaxCount venga mantenuta invariata), il cluster verrà configurato rimuovendo gli ultimi N nodi statici dei nodi statici `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>` e il sistema chiuderà le istanze Amazon corrispondenti. EC2 Inoltre, per rispettare la MaxCount capacità della risorsa di elaborazione, la configurazione del cluster viene aggiornata estendendo il numero di nodi dinamici per colmare il divario. `MaxCount - new_MinCount: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]` In questo caso, trattandosi di nodi dinamici, non verranno lanciate nuove EC2 istanze Amazon a meno che lo scheduler non abbia lavori in sospeso sui nuovi nodi.

- Stato iniziale: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Aggiornamento -30 su MinCount : MinCount = 70 (MaxCount = 120)

- ```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 80 idle~ queue1-dy-c5xlarge-[1-80]
queue1* up infinite 70 idle queue1-st-c5xlarge-[1-70]
```

Se  $\text{MinCount} < \text{MaxCount}$ , in diminuzione  $\text{MinCount}$  e  $\text{MaxCount}$  della stessa quantità  $N$ , il cluster verrà configurato rimuovendo gli ultimi  $N$  nodi statici `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] e il sistema interromperà le istanze Amazon EC2 corrispondenti.

Inoltre, non verrà apportata alcuna modifica al numero di nodi dinamici per rispettare il nuovo valore.  $\text{MaxCount}$

- Stato iniziale:  $\text{MinCount} = 100$ ;  $\text{MaxCount} = 150$

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up       infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Aggiornamento -30 su MinCount : $\text{MinCount} = 70$ ($\text{MaxCount} = 120$)

- ```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 80 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 70 idle queue1-st-c5xlarge-[1-70]
```

Se  $\text{MinCount} < \text{MaxCount}$ , diminuendo la  $\text{MaxCount}$  quantità  $N$  (supponendo che  $\text{MinCount}$  venga mantenuta invariata), il cluster verrà configurato rimuovendo gli ultimi  $N$  nodi dinamici `<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>`] e



il sistema interromperà le EC2 istanze Amazon corrispondenti nel caso in cui fossero in esecuzione. Non è previsto alcun impatto sui nodi statici.

- Stato iniziale: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Aggiornamento -30 su MaxCount : MinCount = 100 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 20 idle~ queue1-dy-c5xlarge-[1-20]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

## Impatti sui lavori

In tutti i casi in cui i nodi vengono rimossi e EC2 le istanze Amazon terminate, un job sbatch in esecuzione sui nodi rimossi verrà messo nuovamente in coda, a meno che non vi siano altri nodi che soddisfino i requisiti del processo. In quest'ultimo caso, il job fallisce con lo stato `NODE_FAIL` e scompare dalla coda e deve essere inviato nuovamente manualmente.

Se si prevede di eseguire un aggiornamento per il ridimensionamento del cluster, è possibile impedire l'esecuzione dei lavori nei nodi che verranno rimossi durante l'aggiornamento pianificato. Ciò è possibile impostando i nodi da rimuovere durante la manutenzione. Tieni presente che l'impostazione di un nodo in manutenzione non influirebbe sui lavori che alla fine sono già in esecuzione nel nodo.

Supponiamo che con l'aggiornamento pianificato per il ridimensionamento del cluster si voglia rimuovere il nodo `queue-st-computeresource-[9-10]`. È possibile creare un Slurm prenotazione con il seguente comando

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queueu-st-
computeresource-[9-10]
```

Questo creerà un Slurm prenotazione denominata `maint_for_update` sui nodi `queueu-st-computeresource-[9-10]`. Dal momento in cui viene creata la prenotazione, non è possibile eseguire altri lavori nei nodi `queueu-st-computeresource-[9-10]`. Tieni presente che la prenotazione non impedirà l'assegnazione finale dei lavori sui nodi `queueu-st-computeresource-[9-10]`.

Dopo l'aggiornamento del ridimensionamento del cluster, se Slurm la prenotazione è stata impostata solo sui nodi che sono stati rimossi durante l'aggiornamento di ridimensionamento, la prenotazione di manutenzione verrà eliminata automaticamente. Se invece avessi creato un Slurm prenotazione sui nodi che sono ancora presenti dopo l'aggiornamento del ridimensionamento del cluster, potremmo voler rimuovere la prenotazione di manutenzione sui nodi dopo aver eseguito l'aggiornamento di ridimensionamento, utilizzando il seguente comando

```
sudo -i scontrol delete ReservationName=maint_for_update
```

Per ulteriori dettagli su Slurm per prenotare, consulta il documento ufficiale di SchedMD [qui](#).

## Processo di aggiornamento del cluster in caso di modifiche alla capacità

In caso di modifica della configurazione dello scheduler, durante il processo di aggiornamento del cluster vengono eseguiti i seguenti passaggi:

- Interrompi AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- Genera aggiornamenti Slurm configurazione delle partizioni dalla AWS ParallelCluster configurazione
- Riavvio `slurmctld` (eseguito tramite la ricetta del servizio Chef)
- Controlla `slurmctld` lo stato (`systemctl is-active --quiet slurmctld.service`)
- Reload (Carica di nuovo) Slurm configurazione (`scontrol reconfigure`)
- Avvia `clustermgtd` (`supervisorctl start clustermgtd`)

## Utilizzo dello scheduler AWS Batch (**awsbatch**) con AWS ParallelCluster

### Warning

AWS CodeBuild non è supportato nelle regioni Asia Pacifico (Malesia) (ap-southeast-5) e Asia Pacifico (Tailandia) (ap-southeast-7). Pertanto, ParallelCluster AWS Batch l'integrazione non è supportata in tali regioni.

AWS ParallelCluster supporta anche AWS Batch gli scheduler. I seguenti argomenti descrivono come utilizzare AWS Batch. Per informazioni su AWS Batch, vedere [AWS Batch](#). Per la documentazione, consulta la [Guida AWS Batch per l'utente](#).

### AWS ParallelCluster Comandi CLI per AWS Batch

Quando si utilizza lo `awsbatch` scheduler, i comandi AWS ParallelCluster CLI AWS Batch per vengono installati automaticamente nel nodo AWS ParallelCluster principale. La CLI utilizza le operazioni AWS Batch API e consente le seguenti operazioni:

- Inviare e gestire attività
- Monitorare attività, code e host
- Creare una copia speculare dei comandi del pianificatore tradizionali

### Important

AWS ParallelCluster non supporta i lavori GPU per. AWS Batch Per ulteriori informazioni, consulta [GPU jobs](#).

Questa CLI è distribuita come pacchetto separato. Per ulteriori informazioni, consulta [Supporto Scheduler](#).

### Argomenti

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)

- [awsbqueues](#)
- [awsbhosts](#)

## awsbsub

Invia i lavori alla coda dei lavori del cluster.

```
awsbsub [-h] [-jn JOB_NAME] [-c CLUSTER] [-cf] [-w WORKING_DIR]
 [-pw PARENT_WORKING_DIR] [-if INPUT_FILE] [-p VCPUS] [-m MEMORY]
 [-e ENV] [-eb ENV_DENYLIST] [-r RETRY_ATTEMPTS] [-t TIMEOUT]
 [-n NODES] [-a ARRAY_SIZE] [-d DEPENDS_ON]
 [command] [arguments [arguments ...]]
```

### Important

AWS ParallelCluster non supporta i lavori GPU per. AWS Batch Per ulteriori informazioni, consulta [GPU jobs](#).

Argomenti posizionali

### ***command***

Invia il lavoro (il comando specificato deve essere disponibile nelle istanze di calcolo) o il nome del file da trasferire. Consulta anche `--command-file`.

### **arguments**

(Facoltativo) Specifica argomenti per il comando o il file di comando.

Argomenti designati

**-jn *JOB\_NAME*, --job-name *JOB\_NAME***

I nomi del processo. Il primo carattere deve essere una lettera o un numero. Il nome del lavoro può contenere lettere (sia maiuscole che minuscole), numeri, trattini e caratteri di sottolineatura e avere una lunghezza massima di 128 caratteri.

**-c *CLUSTER*, --cluster *CLUSTER***

Specifica il cluster da utilizzare.

**-cf, --command-file**

Indica che il comando è un file da trasferire nelle istanze di calcolo.

Impostazione predefinita: False

**-w *WORKING\_DIR*, --working-dir *WORKING\_DIR***

Specifica la cartella da utilizzare come directory di lavoro del processo. Se non viene specificata una directory di lavoro, il processo viene eseguito nella job-*<AWS\_BATCH\_JOB\_ID>* sottocartella della home directory dell'utente. Puoi usare questo parametro o il parametro `--parent-working-dir`.

**-pw *PARENT\_WORKING\_DIR*, --parent-working-dir *PARENT\_WORKING\_DIR***

Specifica la cartella principale della directory di lavoro del lavoro. Se non viene specificata una directory di lavoro principale, per impostazione predefinita è la home directory dell'utente. Una sottocartella denominata job-*<AWS\_BATCH\_JOB\_ID>* viene creata nella directory di lavoro padre. Puoi usare questo parametro o il parametro `--working-dir`.

**-if *INPUT\_FILE*, --input-file *INPUT\_FILE***

Specifica il file da trasferire alle istanze di calcolo, nella directory di lavoro del processo. È possibile specificare più parametri di file di input.

**-p *VCPUS*, --vcpus *VCPUS***

Specifica il numero di v CPUs da riservare per il contenitore. Se usato insieme a `-nodes`, identifica il numero di v CPUs per ogni nodo.

Impostazione predefinita: 1

**-m *MEMORY*, --memory *MEMORY***

Specifica il limite di memoria fisico (in MiB) da fornire per il processo. Se il lavoro tenta di superare il limite di memoria specificato qui, il lavoro viene terminato.

Impostazione predefinita: 128

**-e *ENV*, --env *ENV***

Specifica un elenco separato da virgola di nomi delle variabili di ambiente da esportare nell'ambiente dei processi. Per esportare tutte le variabili di ambiente, specifica "all". Tieni presente che un elenco di «tutte» le variabili di ambiente non include quelle elencate nel `--env-blacklist` parametro o le variabili che iniziano con il `AWS_*` prefisso `PCLUSTER_*` o.

**-e *ENV\_DENYLIST*, --env-blacklist *ENV\_DENYLIST***

Specifica un elenco separato da virgole di nomi di variabili di ambiente da non esportare nell'ambiente dei processi. Per impostazione predefinita, HOME, PWD, USER, PATH, LD\_LIBRARY\_PATH, TERM e TERMCAP non vengono esportate.

**-r *RETRY\_ATTEMPTS*, --retry-attempts *RETRY\_ATTEMPTS***

Specifica il numero di volte in cui un lavoro deve essere spostato. RUNNABLE Puoi specificare da 1 a 10 tentativi. Se il valore dei tentativi è maggiore di 1, il processo viene riprovato se fallisce, finché non passa a uno RUNNABLE stato per il numero di volte specificato.

Impostazione predefinita: 1

**-t *TIMEOUT*, --timeout *TIMEOUT***

Specifica la durata in secondi (misurata in base al startedAt timestamp del tentativo di lavoro) dopo la quale AWS Batch termina il lavoro se non è stato completato. Il valore di timeout deve essere almeno di 60 secondi.

**-n *NODES*, --nodes *NODES***

Specifica il numero di nodi da prenotare per il processo. Specificate un valore per questo parametro per abilitare l'invio parallelo multinodo.

 Note

Quando il [CapacityType](#) parametro [Scheduler/AwsBatchQueues](#) è impostato su SPOT, i lavori paralleli multinodo non sono supportati. Inoltre, nel tuo account deve essere presente un ruolo `AWSServiceRoleForEC2Spot` collegato al servizio. Puoi creare questo ruolo con il seguente comando: AWS CLI

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Per ulteriori informazioni, consulta il [ruolo collegato ai servizi per le richieste di istanze Spot](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

**-a *ARRAY\_SIZE*, --array-size *ARRAY\_SIZE***

Indica le dimensioni dell'array. Puoi specificare un valore compreso tra 2 e 10.000. Se specifichi proprietà dell'array per un processo, diventa un processo in array.

**-d *DEPENDS\_ON*, --depends-on *DEPENDS\_ON***

Specifica un elenco separato da punti e virgola di dipendenze per un processo. Un processo può dipendere da un massimo di 20 processi. Puoi specificare una dipendenza dal SEQUENTIAL tipo senza specificare un ID di lavoro per i job di array. Una dipendenza sequenziale consente a ogni processo in array figlio di terminare sequenzialmente, partendo dall'indice 0. Puoi anche specificare una dipendenza tipo N\_TO\_N con un ID processo per processi in array. Una dipendenza N\_TO\_N significa che ogni figlio nell'indice di questo processo deve attendere il completamento del figlio nell'indice corrispondente di ciascuna dipendenza prima di iniziare. La sintassi di questo parametro è «jobID=<*stringstring*

**awsbstat**

Mostra i processi che vengono inviati nella coda di processi del cluster.

```
awsbstat [-h] [-c CLUSTER] [-s STATUS] [-e] [-d] [job_ids [job_ids ...]]
```

Argomenti posizionali

***job\_ids***

Specifica l'elenco dei lavori separati da spazi da IDs mostrare nell'output. Se il lavoro è un array di attività, vengono visualizzate tutte le attività figlio. Se è richiesto un singolo processo, viene visualizzato in una versione dettagliata.

Argomenti designati

**-c *CLUSTER*, --cluster *CLUSTER***

Indica il cluster da utilizzare.

**-s *STATUS*, --status *STATUS***

Specifica un elenco separato da virgole di stati del processo da includere. Lo stato del processo predefinito è "active". I valori accettati sono: SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING, SUCCEEDED, FAILED e ALL.

Impostazione predefinita: "SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING"

**-e, --expand-children**

Esponde i processi con figli (array e parallelo a più nodi).

Impostazione predefinita: False

**-d, --details**

Mostra i dettagli dei processi.

Impostazione predefinita: False

**awsbout**

Mostra l'output di un determinato processo.

```
awsbout [-h] [-c CLUSTER] [-hd HEAD] [-t TAIL] [-s] [-sp STREAM_PERIOD] job_id
```

Argomenti posizionali

***job\_id***

Specifica l'ID processo.

Argomenti designati

**-c *CLUSTER*, --cluster *CLUSTER***

Indica il cluster da utilizzare.

**-hd *HEAD*, --head *HEAD***

Ottiene le prime *HEAD* righe dell'output del lavoro.

**-t *TAIL*, --tail *TAIL***

Ottiene le ultime righe <tail> dell'output del processo.

**-s, --stream**

Ottiene l'output del processo, quindi attende che venga generato output aggiuntivo. Questo argomento può essere utilizzato insieme a -tail per iniziare dalle ultime righe <tail> dell'output del processo.

Impostazione predefinita: False



**-sp *STREAM\_PERIOD*, --stream-period *STREAM\_PERIOD***

Imposta il periodo di streaming.

Impostazione predefinita: 5

## awsbkill

Annulla o termina i processi inviati nel cluster.

```
awsbkill [-h] [-c CLUSTER] [-r REASON] job_ids [job_ids ...]
```

Argomenti posizionali

### *job\_ids*

Specificate l'elenco separato da spazi dei lavori da annullare o IDs terminare.

Argomenti designati

**-c *CLUSTER*, --cluster *CLUSTER***

Indica il nome del cluster da utilizzare.

**-r *REASON*, --reason *REASON***

Indica il messaggio da collegare a un processo, spiegando il motivo per annullarlo.

Impostazione predefinita:»Terminated by the user”

## awsbqueues

Mostra la coda dei processi associata al cluster.

```
awsbqueues [-h] [-c CLUSTER] [-d] [job_queues [job_queues ...]]
```

Argomenti posizionali

### *job\_queues*

Specifica l'elenco separato da spazi di nomi delle code da visualizzare. Se è richiesta una singola coda, viene mostrata in una versione dettagliata.

## Argomenti denominati

**-c *CLUSTER*, --cluster *CLUSTER***

Specifica il nome del cluster da utilizzare.

**-d, --details**

Indica se visualizzare i dettagli delle code.

Impostazione predefinita: False

## awsbhosts

Mostra gli host che appartengono all'ambiente di calcolo del cluster.

```
awsbhosts [-h] [-c CLUSTER] [-d] [instance_ids [instance_ids ...]]
```

## Argomenti posizionali

***instance\_ids***

Specifica un elenco di istanze separate da spazi. IDs Se è richiesta un'istanza singola, viene mostrata in una versione dettagliata.

## Argomenti designati

**-c *CLUSTER*, --cluster *CLUSTER***

Specifica il nome del cluster da utilizzare.

**-d, --details**

Indica se mostrare i dettagli degli host.

Impostazione predefinita: False

## Archiviazione condivisa

AWS ParallelCluster [supporta l'utilizzo di volumi di storage condivisi Amazon EBS, FSx per ONTAP e FSx OpenZFS, i file system di storage condivisi Amazon EFS e FSx per Lustre o File Caches](#). Ti

consigliamo di seguire le linee guida sull'[affidabilità del framework AWS ben architettato e di eseguire il backup di volumi e file system](#).

Scegliete un sistema di storage che soddisfi i requisiti di I/O delle applicazioni HPC. È possibile ottimizzare ogni file system in base al caso d'uso specifico. Per ulteriori informazioni, consulta la [panoramica delle opzioni di archiviazione](#).

I volumi Amazon EBS sono collegati al nodo principale e condivisi con i nodi di elaborazione tramite NFS. Questa opzione può essere conveniente, ma le prestazioni dipendono dalle risorse del nodo principale in quanto le esigenze di storage sono scalabili. Questo può diventare un collo di bottiglia man mano che vengono aggiunti più nodi di elaborazione al cluster e la richiesta di throughput aumenta.

I file system Amazon EFS si adattano al variare delle esigenze di storage. Puoi configurare questi file system per una varietà di casi d'uso. Usa i file system Amazon EFS per eseguire applicazioni parallelizzate e sensibili alla latenza sul tuo cluster.

FSx i file system for Lustre possono elaborare enormi set di dati con velocità effettiva fino a centinaia di gigabyte al secondo, milioni di IOPS e latenze inferiori al millisecondo. Utilizzatelo FSx per i file system Lustre per ambienti di elaborazione esigenti ad alte prestazioni.

In [Sezione SharedStorage](#), puoi definire lo storage esterno o AWS ParallelCluster gestito:

- Lo storage esterno si riferisce a un volume o file system esistente gestito dall'utente. AWS ParallelCluster non crea o elimina questo spazio di archiviazione.
- Lo storage gestito si riferisce a un volume o a un file system che AWS ParallelCluster è stato creato e può eliminare.

## Archiviazione esterna

È possibile AWS ParallelCluster configurare il collegamento di storage esterno al cluster quando il cluster viene creato o aggiornato. Allo stesso modo, è possibile configurarlo per scollegare lo storage esterno dal cluster quando il cluster viene eliminato o aggiornato. I dati vengono conservati e possono essere utilizzati per l'archiviazione condivisa permanente a lungo termine al di fuori del ciclo di vita del cluster.

### Note

Le versioni AWS ParallelCluster precedenti alla 3.8 non consentono il montaggio di file system gestiti esternamente. /home A partire dalla versione 3.8, AWS ParallelCluster

consente di utilizzarlo /home come punto di montaggio per un file system gestito esterno. È possibile montare un file system gestito esternamente su /home specificando /home come valore il parametro sotto. [MountDirSezione SharedStorage](#)

Amazon File Cache non è adatta all'uso come /home directory di sistema e pertanto al momento non è supportata per il montaggio/home.

Quando si specifica una /home directory sotto l'opzione di configurazione, [Sezione SharedStorage](#) l'opzione di [SharedStorageType](#) configurazione verrà sovrascritta, il che significa che verranno utilizzate al suo posto le impostazioni riportate di [Sezione SharedStorage](#) seguito.

Quando si monta un filesystem esterno /home nella directory, AWS ParallelCluster copia il /home contenuto del nodo principale sul filesystem esterno, senza sovrascrivere i file esistenti sulla memoria esterna. Ciò include il trasferimento della chiave SSH del cluster per l'utente predefinito, se è assente sul filesystem esterno. Per ulteriori informazioni, fare riferimento a. [AWS ParallelCluster considerazioni sullo storage condiviso](#)

## AWS ParallelCluster archiviazione gestita

AWS ParallelCluster per impostazione predefinita nella configurazione, lo storage gestito dipende dal ciclo di vita del cluster. Per impostazione predefinita, il parametro di SharedStorage DeletionPolicy configurazione è impostato Delete su.

Per impostazione predefinita, un file system o volume AWS ParallelCluster gestito e i relativi dati vengono eliminati se si verifica una delle seguenti condizioni.

- Il cluster viene eliminato.
- Si modifica la configurazione dello storage condiviso gestitoName.
- Si rimuove lo storage condiviso gestito dalla configurazione.

Imposta su DeletionPolicy Retain per rendere persistente il file system o il volume e i dati condivisi gestiti. Si consiglia di eseguire regolarmente il backup dei dati per evitare la perdita di dati. È possibile [AWS Backup](#) utilizzarlo per gestire centralmente i backup per tutte le opzioni di archiviazione.

È possibile rimuovere la dipendenza dal ciclo di vita con le impostazioni di configurazione. Per ulteriori informazioni, consulta [Convertire lo storage AWS ParallelCluster gestito in storage esterno](#).

Per informazioni sulle quote di storage condivise, vedere. [Quote per l'archiviazione condivisa](#)

Per ulteriori informazioni sullo storage condiviso e sul passaggio a nuove AWS ParallelCluster versioni, vedere [Migliori pratiche: spostare un cluster in una nuova versione AWS ParallelCluster secondaria o patch](#).

È possibile AWS ParallelCluster configurare il collegamento di storage esterno al cluster quando il cluster viene creato o aggiornato. Allo stesso modo, è possibile configurarlo per scollegare lo storage esterno dal cluster quando il cluster viene eliminato o aggiornato. I dati vengono conservati e possono essere utilizzati per soluzioni di storage condiviso permanenti a lungo termine indipendenti dal ciclo di vita del cluster.

Per impostazione predefinita, lo storage gestito dipende dal ciclo di vita del cluster. È possibile rimuovere la dipendenza con le impostazioni di configurazione descritte in [Convertire lo storage AWS ParallelCluster gestito in storage esterno](#)

Con impostazioni specifiche, puoi ottimizzare ciascuna delle soluzioni di archiviazione supportate per i tuoi casi d'uso.

Per le quote di archiviazione condivise, vedere [Quote per l'archiviazione condivisa](#).

Per ulteriori informazioni sullo storage condiviso e sul passaggio a nuove AWS ParallelCluster versioni, vedere [Migliori pratiche: spostare un cluster in una nuova versione AWS ParallelCluster secondaria o patch](#).

I seguenti argomenti descrivono come configurare lo storage condiviso per ogni servizio di storage che lo AWS ParallelCluster supporta.

#### Argomenti

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx per Lustre](#)
- [Configura FSx lo storage condiviso per ONTAP, FSx OpenZFS e File Cache](#)
- [Utilizzo dello storage condiviso in AWS ParallelCluster](#)
- [Quote per l'archiviazione condivisa](#)

## Amazon Elastic Block Store

Per utilizzare un volume Amazon EBS esterno esistente per lo storage permanente a lungo termine indipendente dal ciclo di vita del cluster, specifica [EbsSettings/VolumeId](#).

Se non lo specifichi [VolumeId](#), per impostazione predefinita, AWS ParallelCluster crea un volume EBS gestito dal [EbsSettings](#) momento della creazione del cluster. AWS ParallelCluster elimina anche il volume e i dati quando il cluster viene eliminato o il volume viene rimosso dalla configurazione del cluster.

Per un volume EBS AWS ParallelCluster gestito, è possibile utilizzare [EbsSettings/DeletionPolicy](#) AWS ParallelCluster per Delete indicare o Snapshot il volume quando il cluster viene eliminato o quando il volume viene rimosso dalla configurazione del cluster. Retain Per impostazione predefinita, DeletionPolicy è impostato su Delete.

#### Warning

Per lo storage condiviso AWS ParallelCluster gestito, DeletionPolicy è impostato su come impostazione Delete predefinita.

Ciò significa che, se si verifica una delle seguenti condizioni, un volume gestito e i relativi dati vengono eliminati:

- Si elimina il cluster.
- Si modifica la configurazione dell'archiviazione condivisa gestita [SharedStorage/Name](#).
- Si rimuove lo storage condiviso gestito dalla configurazione.

Si consiglia di eseguire regolarmente il backup dello storage condiviso con istantanee per evitare la perdita di dati. Per ulteriori informazioni sugli snapshot di Amazon EBS, consulta gli snapshot di [Amazon EBS nella](#) Amazon Elastic Compute Cloud User Guide for Linux Instances. Per informazioni su come gestire i backup dei dati in tutti gli ambienti Servizi AWS, consulta [AWS Backup](#) nella AWS Backup Developer Guide.

## Amazon Elastic File System

Per utilizzare un file system Amazon EFS esterno esistente per lo storage permanente a lungo termine al di fuori del ciclo di vita del cluster, specifica [EfsSettings/FileSystemId](#), per impostazione predefinita, AWS ParallelCluster crea un file system Amazon EFS gestito dal [EfsSettings](#) momento in cui crea il cluster. AWS ParallelCluster elimina anche il file system e i dati quando il cluster viene eliminato o quando il file system viene rimosso dalla configurazione del cluster.

Per un file system Amazon EFS AWS ParallelCluster gestito, puoi usare [EfsSettings/DeletionPolicy](#) AWS ParallelCluster per Delete indicare quando il cluster viene eliminato o quando il file system viene rimosso dalla configurazione del cluster. Retain Per impostazione predefinita, DeletionPolicy è impostato su Delete.

#### Warning

Per lo storage condiviso AWS ParallelCluster gestito, DeletionPolicy è impostato su come impostazione Delete predefinita.

Ciò significa che, se si verifica una delle seguenti condizioni, un file system gestito e i relativi dati vengono eliminati:

- Si elimina il cluster.
- Si modifica la configurazione dell'archiviazione condivisa gestita [SharedStorage/Name](#).
- Si rimuove lo storage condiviso gestito dalla configurazione.

Si consiglia di eseguire regolarmente il backup dello storage condiviso per evitare la perdita di dati. Per ulteriori informazioni su come eseguire il backup di singoli volumi Amazon EFS, consulta [Backup dei file system Amazon EFS](#) nella Amazon Elastic File System User Guide. Per informazioni su come gestire i backup dei dati in tutti gli ambienti Servizi AWS, consulta [AWS Backup](#) nella AWS Backup Developer Guide.

## Amazon FSx per Lustre

Per utilizzare un file system esterno esistente FSx per Lustre per l'archiviazione permanente a lungo termine al di fuori del ciclo di vita del cluster, specificare [FsxLustreSettings/FileSystemId](#)

Se non si specifica [FsxLustreSettings/FileSystemId](#), per impostazione predefinita, AWS ParallelCluster crea un file system gestito FSx per Lustre dal [FsxLustreSettings](#) momento della creazione del cluster. AWS ParallelCluster elimina anche il file system e i dati quando il cluster viene eliminato o quando il file system viene rimosso dalla configurazione del cluster.

Per un file system AWS ParallelCluster gestito FSx per Lustre, è possibile utilizzare [FsxLustreSettings/DeletionPolicy](#) per AWS ParallelCluster fornire istruzioni Delete o Retain il file system quando il cluster viene eliminato o quando il file system viene rimosso dalla configurazione del cluster. Per impostazione predefinita, DeletionPolicy è impostato su Delete.

### Warning

Per l'archiviazione AWS ParallelCluster condivisa gestita, `DeletionPolicy` è impostata su `Delete` default.

Ciò significa che, se si verifica una delle seguenti condizioni, un file system gestito e i relativi dati vengono eliminati:

- Si elimina il cluster.
- Si modifica la configurazione dell'archiviazione condivisa gestita [SharedStorage/Name](#).
- Si rimuove lo storage condiviso gestito dalla configurazione.

Si consiglia di eseguire regolarmente il backup dello storage condiviso per evitare la perdita di dati. È possibile definire i backup nel cluster con [SharedStorageFsxLustreSettings/AutomaticBackupRetentionDayseDailyAutomaticBackupStartTime](#). Per informazioni su come gestire i backup dei dati in tutti gli ambienti Servizi AWS, consulta [AWS Backup](#) nella AWS Backup Developer Guide.

## Configura FSx lo storage condiviso per ONTAP, FSx OpenZFS e File Cache

FSx Per ONTAP, FSx OpenZFS e File Cache, puoi usare [FsxOntapSettings/VolumeId](#), [FsxOpenZfsSettings](#)/e [FileCacheSettings/FileCacheId](#) per specificare il montaggio di un volume esterno esistente o di File Cache per il tuo cluster. [VolumeId](#)

AWS ParallelCluster lo storage condiviso gestito non è supportato FSx per ONTAP, OpenZFS e FSx File Cache.

## Utilizzo dello storage condiviso in AWS ParallelCluster

Nelle sezioni seguenti, imparerai a utilizzare AWS ParallelCluster e condividere lo storage, incluse considerazioni sullo storage condiviso e come convertire lo storage gestito in storage esterno.

### Argomenti

- [AWS ParallelCluster considerazioni sullo storage condiviso](#)
- [Convertire lo storage AWS ParallelCluster gestito in storage esterno](#)



## AWS ParallelCluster considerazioni sullo storage condiviso

Considerate quanto segue quando utilizzate lo storage condiviso in AWS ParallelCluster.

- Esegui il backup dei dati del file system con [AWS Backup](#) un altro metodo per gestire i backup per tutti i tuoi sistemi di storage.
- Per aggiungere spazio di archiviazione condiviso, aggiungi una sezione di archiviazione condivisa al file di configurazione e crei o aggiorni il cluster.
- Per rimuovere lo storage condiviso, rimuovi la sezione di archiviazione condivisa dal file di configurazione e aggiorni il cluster.
- Per sostituire lo storage condiviso AWS ParallelCluster gestito esistente con uno nuovo storage gestito, modifica il valore di [SharedStorage/Name](#) e aggiorna il cluster.

### Warning

Per impostazione predefinita, lo storage AWS ParallelCluster gestito e i dati esistenti vengono eliminati quando si esegue l'aggiornamento del cluster con un nuovo Name parametro. Se devi modificare Name e conservare i dati di storage condiviso gestito esistenti, assicurati di impostarli Retain o di DeletionPolicy eseguire il backup dei dati prima di aggiornare il cluster.

- Se non esegui il backup dei dati di storage AWS ParallelCluster gestito e lo DeletionPolicy è Delete, i dati vengono eliminati quando il cluster viene eliminato o quando lo storage gestito viene rimosso dalla configurazione del cluster e il cluster viene aggiornato.
- Se non esegui il backup dei dati di storage AWS ParallelCluster gestito e lo DeletionPolicy è Retain, il file system viene scollegato prima dell'eliminazione del cluster e può essere ricollegato a un altro cluster come file system esterno. I tuoi dati vengono preservati.
- Se lo storage AWS ParallelCluster gestito viene rimosso dalla configurazione del cluster e lo DeletionPolicy è Retain, può essere ricollegato al cluster come file system esterno mantenendo i dati del cluster conservati.
- A partire dalla AWS ParallelCluster versione 3.4.0, puoi migliorare la sicurezza per i montaggi del file system Amazon EFS configurando [SharedStorage/EfsSettings/EncryptionInTransit](#) le impostazioni. [IamAuthorization](#)
- Quando si monta un filesystem esterno nella directory /home, le AWS ParallelCluster copie del contenuto della directory «head node» s /home directory to the external filesystem. It copies existing data in the /home directory without overwriting existing files or directories on the external

storage. This includes the cluster's SSH key for the default user in case it does not already exist on the external filesystem. Consequently all other clusters that mount the same external filesystem to their respective `/home` avranno anche la stessa chiave SSH per l'utente predefinito del cluster.

- In un ambiente multicluster che monta lo stesso file system esterno nelle directory `/home` dei cluster, le chiavi SSH che garantiscono l'accesso ai nodi di calcolo, create sul nodo principale da AWS ParallelCluster, vengono generate una sola volta quando il primo cluster monta il filesystem esterno su `/home`. Tutti gli altri cluster utilizzano la stessa chiave SSH. Di conseguenza, chiunque possieda la chiave SSH per l'utente predefinito di questi cluster condivisi può accedere a qualsiasi cluster. Tutti i nodi di calcolo consentono le connessioni utilizzando la chiave generata inizialmente.

## Convertire lo storage AWS ParallelCluster gestito in storage esterno

Scopri come convertire lo storage AWS ParallelCluster gestito in storage esterno.

Le procedure si basano sul seguente frammento di file di configurazione di esempio.

```
...
- MountDir: /fsx
 Name: fsx
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeletionPolicy: Delete
...
```

## Convertire lo storage AWS ParallelCluster gestito in uno storage esterno

1. Imposta `DeletionPolicy` to `Retain` nel file di configurazione del cluster.

```
...
- MountDir: /fsx
 Name: fsx
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeletionPolicy: Retain
...
```

2. Per impostare la `DeletionPolicy` modifica, esegui il comando seguente.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

3. Rimuovi la SharedStorage sezione dal file di configurazione del cluster.

```
...
...
```

4. Per cambiare il file gestito SharedStorage in esterno SharedStorage e scollegarlo dal cluster, esegui il comando seguente.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. L'archiviazione condivisa è ora esterna e distaccata dal cluster.
6. Per collegare il file system esterno al cluster originale o a un altro cluster, segui questi passaggi.
  - a. Ottieni l'ID del file system FSx for Lustre.

- i. Per utilizzare il comando, AWS CLI esegui il seguente comando e trova il file system con un nome che includa il nome del cluster originale e annota l'ID del file system.

```
aws fsx describe-file-systems
```

- ii. Per utilizzare il Console di gestione AWS, accedi e vai a <https://console.aws.amazon.com/fsx/>. Nell'elenco dei file system, individua il file system con un nome che includa il nome del cluster originale e annota l'ID del file system.
- b. Aggiorna le regole del gruppo di sicurezza del file system per consentire l'accesso da e verso le sottoreti del file system e del cluster. Puoi trovare il nome e l'ID del gruppo di sicurezza del file system nella FSx console Amazon.

Aggiungi regole al gruppo di sicurezza del file system che consentono il traffico TCP in entrata e in uscita da e verso il nodo principale e gli intervalli o i prefissi CIDR IP del nodo di calcolo. Specificate le porte TCP 988, 1021, 1022 e 1023 per il traffico TCP in entrata e in uscita.

Per ulteriori informazioni, consulta

[SharedStorage/FsxLustreSettings/FileSystemId](#) e [Creazione, configurazione ed eliminazione di gruppi di sicurezza per Amazon EC2](#) nella Guida per l'AWS Command Line Interface utente per la versione 2.

- c. Aggiungi la SharedStorage sezione alla configurazione del cluster.

```
...
- MountDir: /fsx
 Name: fsx-external
 StorageType: FsxLustre
 FsxLustreSettings:
 FileSystemId: fs-02e5b4b4abd62d51c
...
```

- d. Per aggiungere lo storage condiviso esterno al cluster, esegui il comando seguente.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

## Quote per l'archiviazione condivisa

Configura SharedStorage il cluster per montare lo storage di file condiviso esistente e creare un nuovo archivio di file condiviso in base alle quote elencate nella tabella seguente.

Le quote di archiviazione dei file montate per ogni cluster

| Tipo di archiviazione condivisa dei file | AWS ParallelCluster archiviazione gestita | archiviazione esterna | Quota netta totale |
|------------------------------------------|-------------------------------------------|-----------------------|--------------------|
| Amazon EBS                               | 5                                         | 5                     | 5                  |
| INCURSIONE                               | 1                                         | 0                     | 1                  |
| Amazon EFS                               | 1                                         | 20                    | 21                 |
| FSx Amazon†                              | 1 FSx per Lustre                          | 20                    | 21                 |

### Note

Questa tabella delle quote è stata aggiunta nella AWS ParallelCluster versione 3.2.0.

† supporta AWS ParallelCluster solo il montaggio dei sistemi Amazon FSx for NetApp ONTAP, Amazon FSx for OpenZFS e File Cache esistenti. Non supporta la creazione di nuovi sistemi FSx per ONTAP, FSx OpenZFS e File Cache.

#### Note

Se lo usi AWS Batch come scheduler, FSx for Lustre è disponibile solo sul nodo principale del cluster.

Le cache dei file non supportano gli scheduler. AWS Batch

## AWS ParallelCluster risorse e etichettatura

Con AWS ParallelCluster puoi creare tag per tracciare e gestire le tue AWS ParallelCluster risorse. È possibile definire i tag che si desidera AWS CloudFormation creare e propagare a tutte le risorse del cluster nel file [Sezione Tags](#) di configurazione del cluster. È inoltre possibile utilizzare tag generati AWS ParallelCluster automaticamente per tracciare e gestire le risorse.

Quando si crea un cluster, il cluster e le relative risorse vengono etichettati con i tag AWS ParallelCluster e AWS systems definiti in questa sezione.

AWS ParallelCluster applica i tag alle istanze, ai volumi e alle risorse del cluster. Per identificare lo stack del cluster, AWS CloudFormation applica i tag di AWS sistema alle istanze del cluster. Per identificare i modelli di lancio di Amazon EC2 del cluster, Amazon EC2 applica i tag di sistema alle istanze. Puoi utilizzare questi tag per visualizzare e gestire le tue risorse. AWS ParallelCluster

#### Warning

Tutti i AWS ParallelCluster tag sono essenziali e non devono essere modificati per evitare impatti sulla funzionalità del sistema. Per questo motivo, non è possibile modificare i tag AWS di sistema.

Di seguito è riportato un esempio di tag di AWS sistema per una AWS ParallelCluster risorsa.

```
"aws:cloudformation:stack-name"="clustername"
```

Di seguito è riportato un esempio di AWS ParallelCluster tag applicato a una risorsa.

```
"parallelcluster:cluster-name"="clustername"
```

Puoi visualizzare questi tag nella sezione Amazon EC2 del. Console di gestione AWS

## Visualizzare i tag

Completa i seguenti passaggi per visualizzare i tag nella sezione Amazon EC2 del. Console di gestione AWS

### Visualizzare i tag

1. Accedi alla console Amazon EC2 all'indirizzo. <https://console.aws.amazon.com/ec2/>
2. Per visualizzare tutti i tag del cluster, scegli Tag nel riquadro di navigazione.
3. Per visualizzare i tag del cluster per istanza, scegli Istanze nel riquadro di navigazione.
4. Seleziona un'istanza del cluster.
5. Scegli la scheda Gestisci tag nei dettagli dell'istanza e visualizza i tag.
6. Scegli la scheda Archiviazione nei dettagli dell'istanza.
7. Seleziona l'ID del volume.
8. In Volumes, scegli il volume.
9. Scegli la scheda Tag nei dettagli del volume e visualizza i tag.

### AWS ParallelCluster tag di istanza del nodo principale

| Chiave                        | Valore di tag               |
|-------------------------------|-----------------------------|
| parallelcluster:cluster-name  | <i>clustername</i>          |
| Name                          | HeadNode                    |
| aws:ec2launchtemplate:id      | <i>lt-1234567890abcdef0</i> |
| aws:ec2launchtemplate:version | <i>1</i>                    |
| parallelcluster:node-type     | HeadNode                    |
| aws:cloudformation:stack-name | <i>clustername</i>          |

| Chiave                                     | Valore di tag                                                                                                                                |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>aws:cloudformation:logical-id</code> | HeadNode                                                                                                                                     |
| <code>aws:cloudformation:stack-id</code>   | <code>arn:aws:cloudformation: <i>region-id</i> :<i>ACCOUNTID</i> :stack/<i>clustername</i> /1234abcd-12ab-12ab-12ab-1234567890abcdef0</code> |
| <code>parallelcluster:version</code>       | <i>3.15.0</i>                                                                                                                                |

#### AWS ParallelCluster tag del volume principale del nodo principale

| Chiave tag                                | Valore di tag      |
|-------------------------------------------|--------------------|
| <code>parallelcluster:cluster-name</code> | <i>clustername</i> |
| <code>parallelcluster:node-type</code>    | HeadNode           |
| <code>parallelcluster:version</code>      | <i>3.15.0</i>      |

#### AWS ParallelCluster tag di istanza del nodo di calcolo

| Chiave                                             | Valore di tag                |
|----------------------------------------------------|------------------------------|
| <code>parallelcluster:cluster-name</code>          | <i>clustername</i>           |
| <code>parallelcluster:compute-resource-name</code> | <i>compute-resource-name</i> |
| <code>aws:ec2launchtemplate:id</code>              | <i>lt-1234567890abcdef0</i>  |
| <code>aws:ec2launchtemplate:version</code>         | <i>1</i>                     |
| <code>parallelcluster:node-type</code>             | Compute                      |
| <code>parallelcluster:queue-name</code>            | <i>queue-name</i>            |
| <code>parallelcluster:version</code>               | <i>3.15.0</i>                |

## AWS ParallelCluster tag del volume principale del nodo di calcolo

| Chiave tag                                         | Valore di tag                |
|----------------------------------------------------|------------------------------|
| <code>parallelcluster:cluster-name</code>          | <i>clustername</i>           |
| <code>parallelcluster:compute-resource-name</code> | <i>compute-resource-name</i> |
| <code>parallelcluster:node-type</code>             | Compute                      |
| <code>parallelcluster:queue-name</code>            | <i>queue-name</i>            |
| <code>parallelcluster:version</code>               | <i>3.15.0</i>                |

## tag PCUI

| Chiave tag                      | Valore di tag |
|---------------------------------|---------------|
| <code>parallelcluster-ui</code> | true          |

## Monitoraggio AWS ParallelCluster e registri

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle AWS ParallelCluster altre AWS soluzioni esistenti. AWS fornisce i seguenti strumenti di monitoraggio per osservare AWS ParallelCluster, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni contenute nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre



archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).

- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo Account AWS e fornisce i file di log a un bucket Simple Storage Service (Amazon S3) specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consulta la [AWS CloudTrail Guida per l'utente di](#).
- Amazon EventBridge è un servizio di bus eventi senza server che semplifica la connessione delle applicazioni con dati provenienti da una varietà di fonti. EventBridge fornisce un flusso di dati in tempo reale dalle tue applicazioni, applicazioni Software-as-a-Service (SaaS) e AWS servizi e indirizza tali dati verso destinazioni come Lambda. In questo modo puoi monitorare gli eventi che si verificano nei servizi e creare architetture basate su eventi. Per ulteriori informazioni, consulta la [Amazon EventBridge User Guide](#).

## Argomenti

- [Integrazione con Amazon CloudWatch Logs](#)
- [CloudWatch Pannello di controllo Amazon](#)
- [CloudWatch Allarmi Amazon per le metriche dei cluster](#)
- [AWS ParallelCluster rotazione dei log configurata](#)
- [pclusterRegistri CLI](#)
- [Registri di output della EC2 console Amazon](#)
- [Recupera i log di PCUI e AWS ParallelCluster di runtime](#)
- [Recupero e conservazione dei log](#)

## Integrazione con Amazon CloudWatch Logs

Per ulteriori informazioni sui CloudWatch log, consulta la [Amazon CloudWatch Logs User Guide](#). Per configurare l'integrazione CloudWatch dei log, consulta la sezione. [Monitoring](#) Per informazioni su come aggiungere log personalizzati alla CloudWatch configurazione utilizzando `append-config`, consulta [File di configurazione per CloudWatch agenti multipli](#) nella Amazon CloudWatch User Guide.

## Registri del cluster Amazon CloudWatch Logs

Viene creato un gruppo di log per ogni cluster con un nome `/aws/parallelcluster/cluster-name-<timestamp>` (ad esempio, `/aws/parallelcluster/testCluster-202202050215`).

Ogni registro (o set di log se il percorso contiene un\*) su ogni nodo ha un flusso di log denominato `{hostname}.{instance_id}.{logIdentifier}`. (Ad esempio `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`.) I dati di registro vengono inviati CloudWatch dall'[CloudWatch agente](#), che viene eseguito come root su tutte le istanze del cluster.

Una CloudWatch dashboard Amazon viene creata al momento della creazione del cluster. Questa dashboard ti dà la possibilità di esaminare i log archiviati in CloudWatch Logs. Per ulteriori informazioni, consulta [CloudWatch Pannello di controllo Amazon](#).

Questo elenco contiene il percorso `logIdentifier` e il percorso per i flussi di log disponibili per piattaforme, scheduler e nodi.

Stream di log disponibili per piattaforme, scheduler e nodi

| Piattaforme                           | Pianificatori                | Nodi                      | Flussi di log                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------|------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| amazon<br>cappello<br>rosso<br>ubuntu | un<br>batch<br>baracco<br>li | HeadNode                  | autenticatore dcv: <code>/var/log/parallelcluster/pc<br/>luster_dcv_authenticator.log</code><br><br>dcv-ext-authenticator: <code>/var/log/parallelcluster/pc<br/>luster_dcv_connect.log</code><br><br>agente dcv: <code>/var/log/dcv/agent.*.log</code><br><br>sessione dcv: <code>/var/log/dcv/dcv-xsession.*.log</code><br><br>server dcv: <code>/var/log/dcv/server.log</code><br><br>dcv-session-launcher: <code>/var/log/dcv/sessionlauncher.log</code><br><br>Dcv: <code>/var/log/dcv/Xdcv.*.log</code><br><br>cfn-init: <code>/var/log/cfn-init.log</code><br><br>chef-client: <code>/var/log/chef-client.log</code> |
| amazon<br>cappello<br>rosso           | un<br>batch                  | Compute<br>et<br>HeadNode | cloud-init: <code>/var/log/cloud-init.log</code><br><br>supervisori: <code>/var/log/supervisord.log</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Piattaforma                               | Pianificatori | Nodi     | Flussi di log                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------|---------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ubuntu                                    | baracoli      |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| amazon<br>cappello<br>rosso<br><br>ubuntu | baracoli      | Compute  | cloud-init-output: /var/log/cloud-init-output.log<br><br>computemgtd: /var/log/parallelcluster/computemgtd<br><br>slurmd: /var/log/slurmd.log<br><br>slurm_prolog_epilog: /var/log/parallelcluster/slurm_prolog_epilog.log                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| amazon<br>cappello<br>rosso<br><br>ubuntu | baracoli      | HeadNode | sssd: /var/log/sssd/sssd.log<br><br>dominio_sssd_predefinito: /var/log/sssd/sssd_default.log<br><br>pam_ssh_key_generator: /var/log/parallelcluster/pam_ssh_key_generator.log<br><br>clusterstatusmgtd: /var/log/parallelcluster/clusterstatusmgtd<br><br>clustermgtd: /var/log/parallelcluster/clustermgtd<br><br>output della console del computer: /var/log/parallelcluster/compute_console_output<br><br>slurm_curriculum: /var/log/parallelcluster/slurm_resume.log<br><br>slurm_sospendere: /var/log/parallelcluster/slurm_suspend.log<br><br>slurmctld: /var/log/slurmctld.log<br><br>slurm_fleet_status_manager: /var/log/parallelcluster/slurm_fleet_status_manager.log |

| Piattaforme                 | Pianificatori             | Nodi                    | Flussi di log                          |
|-----------------------------|---------------------------|-------------------------|----------------------------------------|
| amazon<br>cappelli<br>rosso | un<br>batch<br>baraccolli | Comput<br>eet<br>HeadNc | messaggi di sistema: /var/log/messages |
| ubuntu                      | awsbatch<br>baraccolli    | Comput<br>eet<br>HeadNc | syslog: /var/log/syslog                |

I lavori nei cluster che utilizzano AWS Batch memorizzano l'output dei lavori che hanno raggiunto uno stato di `RUNNING`, `SUCCEEDED`, o `FAILED` in Logs. CloudWatch Il gruppo di log è `/aws/batch/job` il formato del nome del flusso di log è `jobDefinitionName/default/ecs_task_id` Per impostazione predefinita, questi registri sono impostati per non scadere, ma è possibile modificare il periodo di conservazione. Per ulteriori informazioni, consulta [Change log data retention in CloudWatch Logs](#) nella Amazon CloudWatch Logs User Guide.

## Amazon CloudWatch Logs crea log di immagini

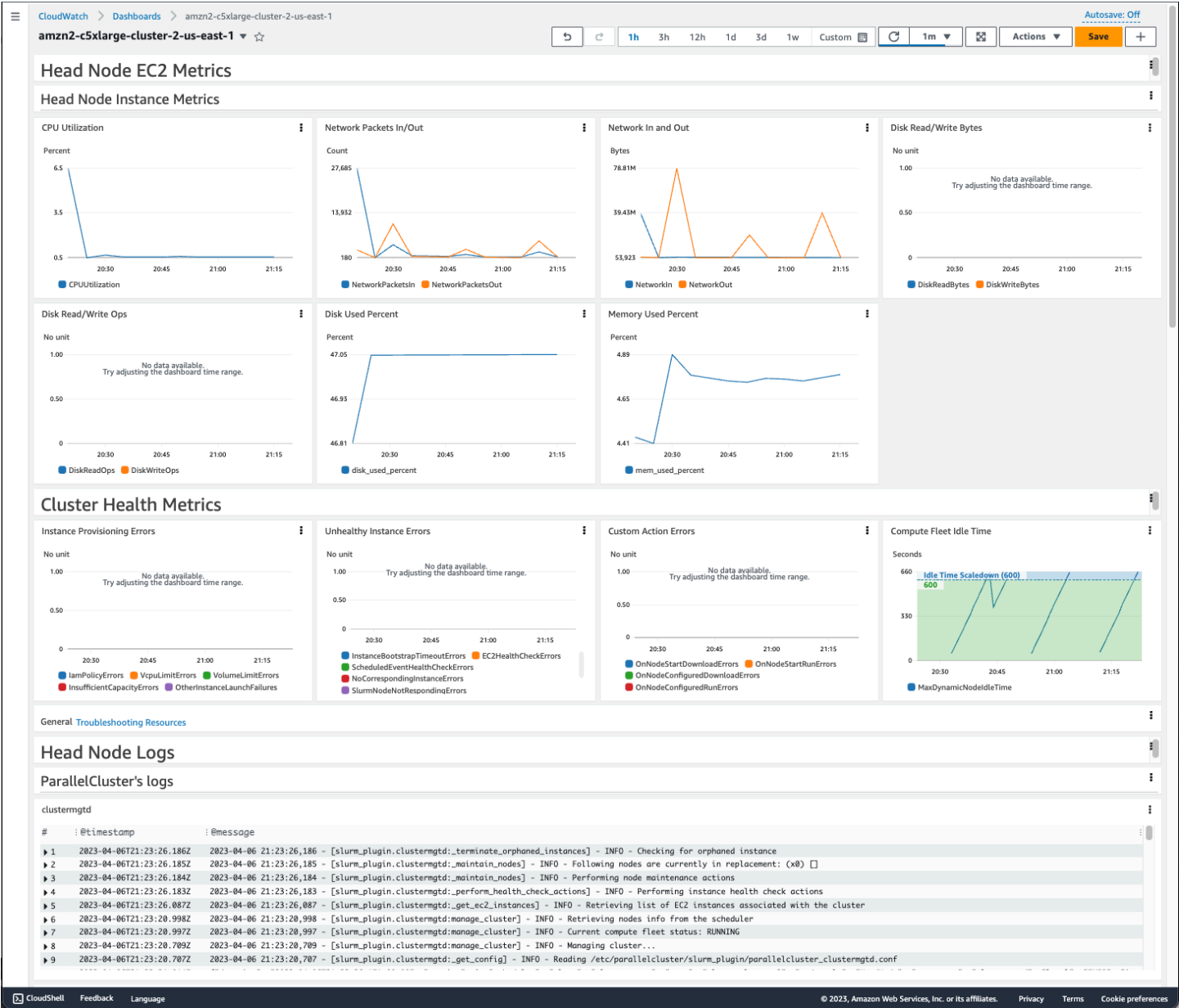
Viene creato un gruppo di log per ogni immagine di build personalizzata con un nome, `/aws/imagebuilder/ParallelClusterImage-<image-id>` Un flusso di log univoco con nome `{pcluster-version}` /1 contiene l'output del processo di creazione dell'immagine.

È possibile accedere ai log utilizzando i comandi `pclusterimage`. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

## CloudWatch Pannello di controllo Amazon

Una CloudWatch dashboard Amazon viene creata quando viene creato un cluster. Ciò semplifica il monitoraggio dei nodi del cluster e la visualizzazione dei log archiviati in Amazon CloudWatch Logs. Il nome della dashboard è `ClusterName-Region ClusterName` è il nome del cluster e indica `Region` Regione AWS il cluster in cui si trova. Puoi accedere alla dashboard nella console o aprendola `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`.

L'immagine seguente mostra un esempio di CloudWatch dashboard per un cluster.



Metriche dell'istanza Head Node

La prima sezione della dashboard mostra i grafici delle EC2 metriche Amazon del nodo principale.

Se il tuo cluster dispone di storage condiviso, la sezione successiva mostra i parametri dello storage condiviso.

Metriche di Cluster Health

Se il tuo cluster utilizza Slurm per la pianificazione, i grafici delle metriche sullo stato del cluster mostrano gli errori dei nodi di calcolo del cluster in tempo reale. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi alle metriche di integrità del cluster](#). Le metriche sullo stato del cluster vengono aggiunte alla dashboard a partire dalla versione 3.6.0. AWS ParallelCluster

## Registri del nodo principale

L'ultima sezione elenca i log del nodo principale raggruppati per log AWS ParallelCluster di, log di Scheduler, log di integrazione Amazon DCV e log di sistema.

Per ulteriori informazioni sui CloudWatch pannelli di controllo di Amazon, consulta [Using Amazon CloudWatch dashboard](#) nella Amazon CloudWatch User Guide.

Se non desideri creare la CloudWatch dashboard di Amazon, puoi disattivarla impostando [Monitoring/Dashboards/CloudWatch/Enabled](#) su `false`.

### Note

Se disabiliti la creazione della CloudWatch dashboard di Amazon, disabiliti anche Amazon CloudWatch `disk_used_percent` e gli `memory_used_percent` allarmi per il tuo cluster. Per ulteriori informazioni, consulta [CloudWatch Allarmi Amazon per le metriche dei cluster](#). Gli `memory_used_percent` allarmi `disk_used_percent` e vengono aggiunti a partire dalla AWS ParallelCluster versione 3.6.

## CloudWatch Allarmi Amazon per le metriche dei cluster

AWS ParallelCluster configura gli CloudWatch allarmi Amazon per monitorare lo stato e l'utilizzo delle risorse del nodo principale. Gli allarmi hanno un nome `cluster-name-HeadNode-metric`, `cluster-name` dov'è il nome del cluster e `metric` identifica la metrica da monitorare.

Accedi agli allarmi nella CloudWatch console selezionando Allarmi nel pannello di navigazione.

Un allarme composito denominato `cluster-name-HeadNode` entra nello ALARM stato in cui si attiva uno qualsiasi dei singoli allarmi del nodo principale.

## Allarmi su disco e memoria

A partire dalla AWS ParallelCluster versione 3.6.0, vengono creati i seguenti CloudWatch allarmi:

- ***cluster-name***-HeadNode-Disk— Monitora la metrica del volume principale.  
disk\_used\_percent Entra nello ALARM stato in cui l'utilizzo del disco è superiore al 90% per 1 punto dati in un periodo di 1 minuto.
- ***cluster-name***-HeadNode-Mem— Monitora la mem\_used\_percent metrica. Entra nello ALARM stato in cui l'utilizzo della memoria è superiore al 90% per 1 punto dati in un periodo di 1 minuto.

Per ulteriori informazioni, consulta le [metriche raccolte dall' CloudWatchagente](#) nella Amazon CloudWatch User Guide.

## Health check e allarmi CPU

A partire dalla AWS ParallelCluster versione 3.8.0, vengono creati i seguenti CloudWatch allarmi:

- ***cluster-name***-HeadNode-Health— Monitora il parametro Amazon StatusCheckFailed EC2. Entra nello ALARM stato quando il valore è maggiore di 0 per 1 punto dati in un periodo di 1 minuto.
- ***cluster-name***-HeadNode-Cpu— Monitora il parametro Amazon CPUUtilization EC2. Immette lo ALARM stato in cui l'utilizzo della CPU è superiore al 90% per 1 punto dati in un periodo di 1 minuto.

## Heartbeat alarm (daemon) di gestione del cluster

A partire dalla AWS ParallelCluster versione 3.15.0, quando la CloudWatch registrazione di Amazon è abilitata e viene utilizzato lo Slurm scheduler, viene creato il seguente allarme:

- ***cluster-name***-HeadNode-ClustermgtdHeartbeat— Monitora la metrica nel ClustermgtdHeartbeat namespace. ParallelCluster L'allarme entra in funzione quando ALARM viene ricevuto meno di 1 battito cardiaco per 10 punti dati consecutivi in un periodo di 1 minuto. I dati mancanti vengono considerati violazioni.

### Note

Tutti gli allarmi vengono ripristinati in modo simmetrico: anche il ripristino è regolato dagli stessi punti dati e dallo stesso periodo di valutazione che attivano l'allarme. Ad esempio, gli allarmi con 1 punto dati vengono ripristinati dopo 1 punto dati valido entro lo stesso periodo

di osservazione, allo stesso modo l'`ClustermgtdHeartbeat` allarme richiede 10 punti dati consecutivi validi (10 minuti) per tornare. OK

#### Note

AWS ParallelCluster non configura le azioni di allarme. Per informazioni su come impostare le azioni di allarme, come l'invio di notifiche, vedi [Azioni di allarme](#). Per ulteriori informazioni sugli CloudWatch allarmi Amazon, consulta [Using Amazon CloudWatch alarms](#) nella Amazon CloudWatch User Guide.

Per la AWS ParallelCluster versione 3.8.0 e successive, disabilita gli allarmi impostando [Monitoring/Alarms/Enabled](#) su nella configurazione del cluster. `false`

Per AWS ParallelCluster le versioni precedenti alla 3.8.0, disabilita gli allarmi impostando [Monitoring//DashboardsCloudWatch/Enabled](#) su nella configurazione del cluster.

`false` Tieni presente che questa impostazione disabilita anche la CloudWatch dashboard di Amazon. Vedi [CloudWatch Pannello di controllo Amazon](#) per ulteriori dettagli.

## AWS ParallelCluster rotazione dei log configurata

Le configurazioni di rotazione dei AWS ParallelCluster log si trovano nei `/etc/logrotate.d/parallelcluster*_log_rotation` file. Quando un registro configurato ruota, il contenuto del registro corrente viene conservato in un unico backup e il registro svuotato riprende la registrazione.

Viene mantenuto un solo backup per ogni registro configurato.

AWS ParallelCluster configura un registro in rapida crescita in modo che ruoti quando raggiunge una dimensione di 50 MB. I log a crescita rapida sono correlati alla scalabilità e Slurm, inclusi `/var/log/parallelcluster/clustermgtd`, e `/var/log/parallelcluster/slurm_resume.log` / `/var/log/slurmctld.log`

AWS ParallelCluster configura un registro a crescita lenta in modo che ruoti quando raggiunge una dimensione di 10 MB.

È possibile visualizzare i log precedenti che vengono conservati per il numero di giorni definito nell'impostazione di configurazione [Logs//CloudWatch](#) del cluster con la registrazione abilitata. [RetentionInDays](#) CloudFormation Controlla le `RetentionInDays` impostazioni per vedere se è necessario aumentare il numero di giorni per il tuo caso d'uso.



## AWS ParallelCluster configura e ruota i seguenti registri:

### log del nodo principale

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cfn-init.log
/var/log/chef-client.log
/var/log/dcv/server.log
/var/log/dcv/sessionlauncher.log
/var/log/dcv/agent.*.log
/var/log/dcv/dcv-xsession.*.log
/var/log/dcv/Xdcv.*.log
/var/log/parallelcluster/pam_ssh_key_generator.log
/var/log/parallelcluster/clustermgtd
/var/log/parallelcluster/clusterstatusmgtd
/var/log/parallelcluster/slurm_fleet_status_manager.log
/var/log/parallelcluster/slurm_resume.log
/var/log/parallelcluster/slurm_suspend.log
/var/log/slurmctld.log
/var/log/slurmdbd.log
/var/log/parallelcluster/compute_console_output.log
```

### Registri dei nodi di calcolo

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log
```

### Registri dei nodi di accesso

```
/var/log/cloud-init.log
/var/log/cloud-init.log
/var/log/cloud-init-output.log
/var/log/supervisord.log
/var/log/parallelcluster/pam_ssh_key_generator.log
```

## pclusterRegistri CLI

La pcluster CLI scrive i registri dei comandi nei file `inpcluster.log.#. /home/user/.parallelcluster/`

Per ogni comando, i log generalmente includono il comando con input, una copia della versione dell'API CLI utilizzata per creare il comando, la risposta e informazioni e messaggi di errore. Per un comando di creazione e compilazione, i log includono anche il file di configurazione, le operazioni di convalida del file di configurazione, il CloudFormation modello e i comandi stack.

È possibile utilizzare questi registri per verificare errori, input, versioni e comandi CLI pcluster. Possono anche fungere da registrazione di quando sono stati eseguiti i comandi.

## Registri di output della EC2 console Amazon

Quando AWS ParallelCluster rileva che un'istanza statica del nodo di calcolo termina in modo imprevisto, tenta di recuperare l'output della EC2 console Amazon dall'istanza del nodo terminata dopo un certo periodo di tempo. In questo modo, se il nodo di elaborazione non è in grado di comunicare con Amazon CloudWatch, dall'output della console potrebbero comunque essere recuperate utili informazioni per la risoluzione dei problemi sul motivo per cui il nodo è terminato. L'output della console viene registrato nel `/var/log/parallelcluster/compute_console_output` registro sul nodo principale. Per ulteriori informazioni sull'output della EC2 console Amazon, consulta l'[output della console di istanza](#) nella Amazon EC2 User Guide for Linux Instances.

Per impostazione predefinita, recupera l'output della console AWS ParallelCluster solo da un sottoinsieme campione di nodi terminati. In questo modo si evita che il nodo principale del cluster venga sovraccaricato da più richieste di output da console causate da un numero elevato di terminazioni. Per impostazione predefinita, AWS ParallelCluster attende 5 minuti tra il rilevamento della terminazione e il recupero dell'output della console per dare ad Amazon il EC2 tempo di recuperare l'output finale della console dai nodi.

Puoi modificare la dimensione del campione e i valori dei parametri del tempo di attesa nel `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` file sul nodo principale.

Questa funzionalità è stata aggiunta nella AWS ParallelCluster versione 3.5.0.

## Parametri di output della EC2 console Amazon

Puoi modificare i valori dei seguenti parametri di output della EC2 console Amazon nel `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` file sul nodo principale.

### **compute\_console\_logging\_enabled**

Per disabilitare la raccolta dei log di output della console, imposta `compute_console_logging_enabled` su `false`. Il valore predefinito è `true`.

Puoi aggiornare questo parametro in qualsiasi momento, senza interrompere la flotta di elaborazione.

### **compute\_console\_logging\_max\_sample\_size**

`compute_console_logging_max_sample_size` imposta il numero massimo di nodi di elaborazione da cui AWS ParallelCluster raccoglie gli output della console ogni volta che rileva una terminazione imprevista. Se questo valore è inferiore a 1, AWS ParallelCluster recupera l'output della console da tutti i nodi terminati. Il valore predefinito è 1.

È possibile aggiornare questo parametro in qualsiasi momento, senza interrompere la flotta di elaborazione.

### **compute\_console\_wait\_time**

`compute_console_wait_time` imposta il tempo, in secondi, che intercorre tra AWS ParallelCluster il rilevamento di un guasto del nodo e la raccolta dell'output della console da quel nodo. Puoi aumentare il tempo di attesa se ritieni che Amazon EC2 abbia bisogno di più tempo per raccogliere l'output finale dal nodo terminato. Il valore predefinito è 300 secondi (5 minuti).

È possibile aggiornare questo parametro in qualsiasi momento, senza interrompere la flotta di elaborazione.

## Recupera i log di PCUI e AWS ParallelCluster di runtime

Scopri come recuperare i log PCUI e di AWS ParallelCluster runtime per la risoluzione dei problemi. Per iniziare, trova i nomi PCUI e dello stack pertinenti. AWS ParallelCluster Usa il nome dello stack per individuare i gruppi di log di installazione. Per finire, esporta i log. Questi registri sono specifici del AWS ParallelCluster runtime. Per i log dei cluster, vedere. [Recupero e conservazione dei log](#)

## Prerequisiti

- AWS CLI È installato.
- L'utente dispone delle credenziali per eseguire AWS CLI i comandi su cui è attivo il PCUI. Account AWS
- Puoi accedere alla CloudWatch console Amazon su cui è attiva la PCUI. Account AWS

## Fase 1: Individua i nomi degli stack pertinenti

Nell'esempio seguente, sostituite il testo evidenziato in rosso con i valori effettivi.

Elenca gli stack, usando il Regione AWS punto in cui hai installato il PCUI:

```
$ aws cloudformation list-stacks --region aws-region-id
```

Annota i nomi degli stack per i seguenti stack:

- Il nome dello stack che ha distribuito il PCUI sul tuo account. Hai inserito questo nome quando hai installato il PCUI; ad esempio, `pcluster-ui`
- Lo AWS ParallelCluster stack a cui è preceduto dal nome dello stack immesso, ad esempio. `pcluster-ui-ParallelClusterApi-ABCD1234EFGH`

## Fase 2: Individua i gruppi di log

Elenca i gruppi di log dello stack PCUI, come mostrato nell'esempio seguente:

```
$ aws cloudformation describe-stack-resources \
 --region aws-region-id \
 --stack-name pcluster-ui \
 --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' &&
(LogicalResourceId == 'ApiGatewayAccessLog' || LogicalResourceId ==
'ParallelClusterUILambdaLogGroup')].PhysicalResourceId" \
 --output text
```

Elenca i gruppi di log dello stack AWS ParallelCluster API, come mostrato nell'esempio seguente:

```
$ aws cloudformation describe-stack-resources \
 --region aws-region-id \
```

```
--stack-name pcluster-ui-ParallelCluster-Api-ABCD1234EFGH \
--query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' && LogicalResourceId
== 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \
--output text
```

Nota gli elenchi dei gruppi di log da utilizzare nel passaggio successivo.

### Fase 3: Esportazione dei log

Utilizza i seguenti passaggi per raccogliere ed esportare i registri:

1. Accedi a Console di gestione AWS, quindi accedi alla CloudWatch console [Amazon](#) su cui è acceso il PCUI. Account AWS
2. Scegli Logs, Logs Insights nel riquadro di navigazione.
3. Seleziona tutti i gruppi di log elencati nel passaggio precedente.
4. Scegli un intervallo di tempo, ad esempio 12 ore.
5. Eseguire la seguente query:

```
$ fields @timestamp, @message
| sort @timestamp desc
| limit 10000
```

6. Scegli Esporta risultati, Scarica tabella (JSON).

### Recupero e conservazione dei log

AWS ParallelCluster crea EC2 metriche Amazon per istanze HeadNode e storage di calcolo. Puoi visualizzare le metriche nei dashboard personalizzati della CloudWatch console. AWS ParallelCluster crea anche flussi di CloudWatch log del cluster in gruppi di log. È possibile visualizzare questi registri nei dashboard personalizzati o nei CloudWatch gruppi di log della console. La sezione Configurazione del cluster di [monitoraggio](#) descrive come modificare i CloudWatch registri del cluster e il dashboard. Per ulteriori informazioni, consultare [Integrazione con Amazon CloudWatch Logs](#) e [CloudWatch Pannello di controllo Amazon](#).

I log sono una risorsa utile per la risoluzione dei problemi. Ad esempio, se si desidera eliminare un cluster in errore, potrebbe essere utile creare prima un archivio dei log del cluster. Segui i passaggi indicati [Registri di archivio](#) per creare un archivio.

#### Argomenti

- [I log del cluster non sono disponibili in CloudWatch](#)
- [Registri di archivio](#)
- [Registri conservati](#)
- [Registri dei nodi terminati](#)

## I log del cluster non sono disponibili in CloudWatch

Se i log del cluster non sono disponibili in CloudWatch, assicurati di non aver sovrascritto la configurazione del AWS ParallelCluster CloudWatch registro quando aggiungi log personalizzati alla configurazione.

Per aggiungere log personalizzati alla CloudWatch configurazione, assicurati di aggiungerli alla configurazione anziché recuperarli e sovrascriverli. Per ulteriori informazioni su *fetch-config* e *append-config*, consulta [File di configurazione per più CloudWatch agenti](#) nella Guida per l'utente. CloudWatch

Per ripristinare la configurazione del AWS ParallelCluster CloudWatch registro, puoi eseguire i seguenti comandi all'interno di un AWS ParallelCluster nodo:

```
$ PLATFORM="$(ohai platform | jq -r ".[]")"
LOG_GROUP_NAME="$(cat /etc/chef/dna.json | jq -r ".cluster.log_group_name")"
SCHEDULER="$(cat /etc/chef/dna.json | jq -r ".cluster.scheduler")"
NODE_ROLE="$(cat /etc/chef/dna.json | jq -r ".cluster.node_type")"
CONFIG_DATA_PATH="/usr/local/etc/cloudwatch_agent_config.json"
/opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/python /usr/local/bin/
write_cloudwatch_agent_json.py --platform $PLATFORM --config $CONFIG_DATA_PATH --log-
group $LOG_GROUP_NAME --scheduler $SCHEDULER --node-role $NODE_ROLE
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
-c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json -s
```

## Registri di archivio

Puoi archiviare i log in Amazon S3 o in un file locale (a seconda `--output-file` del parametro).

### Note

A partire dalla versione AWS ParallelCluster 3.12.0, puoi esportare i log nel bucket predefinito. AWS ParallelCluster In questo caso non è necessario impostare le autorizzazioni del bucket.

 Note

Aggiungi le autorizzazioni alla policy dei bucket di Amazon S3 per concedere l'accesso. CloudWatch Per ulteriori informazioni, consulta [Impostare le autorizzazioni su un bucket Amazon S3](#) nella CloudWatch Logs User Guide.

```
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
 --bucket bucketname --bucket-prefix logs
{
 "url": "https://bucketname.s3.eu-west-1.amazonaws.com/export-log/mycluster-
logs-202109071136.tar.gz?..."
}

use the --output-file parameter to save the logs locally
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
 --bucket bucketname --bucket-prefix logs --output-file /tmp/archive.tar.gz
{
 "path": "/tmp/archive.tar.gz"
}
```

L'archivio contiene CloudWatch i flussi di Amazon Logs e gli eventi CloudFormation dello stack dal nodo principale e dai nodi di calcolo degli ultimi 14 giorni, a meno che non sia specificato esplicitamente nella configurazione o nei parametri del comando. `export-cluster-logs` Il tempo necessario per completare il comando dipende dal numero di nodi nel cluster e dal numero di flussi di log disponibili in Logs. CloudWatch Per ulteriori informazioni sui flussi di log disponibili, vedere. [Integrazione con Amazon CloudWatch Logs](#)

## Registri conservati

A partire dalla versione 3.0.0, AWS ParallelCluster conserva i CloudWatch log per impostazione predefinita quando un cluster viene eliminato. Se desideri eliminare un cluster e conservarne i log, assicurati che [Monitoring/LogsCloudWatch/DeletionPolicy](#) non sia impostato su `None` nella configurazione del cluster. Delete Altrimenti, modifica il valore di questo campo in `Retain` ed esegui il `pcluster update-cluster` comando. Quindi, esegui `pcluster delete-cluster --cluster-name <cluster_name>` per eliminare il cluster, ma conserva il gruppo di log archiviato in Amazon CloudWatch.

## Registri dei nodi terminati

Se un nodo di calcolo statico termina in modo imprevisto e non CloudWatch contiene alcun log, controlla se l'output della console per quel nodo di calcolo AWS ParallelCluster è stato registrato sul nodo principale del registro. `/var/log/parallelcluster/compute_console_output` Per ulteriori informazioni, consulta [Registri chiave per il debug](#).

Se il `/var/log/parallelcluster/compute_console_output` log non è disponibile o non contiene l'output per il nodo, usa il per recuperare l'output della console dal AWS CLI nodo guasto. Accedi al nodo principale del cluster e recupera il nodo guasto `instance-id` dal `/var/log/parallelcluster/slurm_resume.log` file.

Recupera l'output della console utilizzando il seguente comando con: `instance-id`

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Se un nodo di calcolo dinamico si interrompe automaticamente dopo l'avvio e non CloudWatch contiene alcun registro, invia un processo che attivi un'azione di scalabilità del cluster. Attendi che l'istanza fallisca e recupera il log della console dell'istanza.

Accedi al nodo principale del cluster e recupera il nodo di calcolo `instance-id` dal `/var/log/parallelcluster/slurm_resume.log` file.

Per recuperare il registro della console dell'istanza, utilizzate il seguente comando:

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Il log di output della console può aiutarti a eseguire il debug della causa principale di un errore del nodo di calcolo quando il log del nodo di calcolo non è disponibile.

## AWS CloudFormation risorsa personalizzata

A partire dalla AWS ParallelCluster versione 3.6.0, puoi utilizzare una risorsa AWS ParallelCluster CloudFormation personalizzata in uno stack. AWS CloudFormation La risorsa personalizzata è uno AWS ParallelCluster stack ospitato. In questo modo, puoi utilizzarlo CloudFormation per configurare e gestire i tuoi cluster. Ad esempio, è possibile configurare risorse esterne del cluster come rete, storage condiviso e infrastruttura dei gruppi di sicurezza in uno CloudFormation stack. Inoltre, puoi gestire il cluster con un' CloudFormation infrastruttura come pipeline di codice.



Aggiungi una risorsa AWS ParallelCluster personalizzata al tuo CloudFormation modello effettuando le seguenti operazioni:

1. Aggiungi uno stack di provider di risorse personalizzato di proprietà e ospitato da AWS ParallelCluster.
2. Fai riferimento allo stack di provider nel tuo CloudFormation modello come risorsa personalizzata.

Lo stack di provider di risorse personalizzato gestisce e risponde alle richieste. CloudFormation Ad esempio, quando si distribuisce lo CloudFormation stack, si configura e si crea anche un cluster. Per aggiornare un cluster, aggiorni il tuo CloudFormation stack. Elimini un cluster quando elimini lo stack. Per ulteriori informazioni sulle risorse CloudFormation personalizzate, consulta [Risorse personalizzate](#) nella Guida per l'AWS CloudFormation utente.

#### Warning

CloudFormation non rileva la deriva delle risorse personalizzate. Viene utilizzato solo CloudFormation per aggiornare la configurazione del cluster e per eliminare un cluster. È possibile utilizzare la [pcluster](#) CLI o il [AWS ParallelCluster UI](#) per monitorare lo stato del cluster o per aggiornare la flotta di calcolo, ma non è necessario utilizzarli per aggiornare la configurazione del cluster o per eliminare il cluster.

#### Note

Ti consigliamo di aggiungere la [protezione dalla terminazione](#) allo stack per evitare la rimozione accidentale.

## Stack di provider ospitato da AWS ParallelCluster

Lo stack di provider di risorse personalizzato è formattato come mostrato nel seguente frammento di CloudFormation modello:

```
PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 Parameters:
```

```
CustomLambdaRole: # (Optional) RoleARN to override default
AdditionalIamPolicies: # (Optional) comma-separated list of IAM policies to add
TemplateURL: !Sub
- https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.${AWS::URLSuffix}/
parallelcluster/${Version}/templates/custom_resource/cluster.yaml
- { Version: 3.15.0 }
```

## Proprietà:

### Parametri:

#### CustomLambdaRole (opzionale):

Un ruolo personalizzato con autorizzazioni per l'esecuzione AWS Lambda che crea e gestisce il cluster. Per impostazione predefinita, il ruolo utilizza le stesse politiche definite di default nella [AWS ParallelCluster documentazione](#).

#### AdditionalIamPolicies (opzionale):

Un elenco separato da virgole di Amazon Resource Names (ARNs) aggiuntivi di IAM Policy da aggiungere al ruolo utilizzato da Lambda. Viene utilizzato solo se a CustomLambdaRole non è specificato e può essere lasciato vuoto.

Se hai bisogno di policy aggiuntive per il nodo principale, i nodi di calcolo o per l'accesso a un bucket Amazon S3, aggiungile alla CustomLambdaRole proprietà or. AdditionalIamPolicy

Se devi allegare politiche aggiuntive al nodo principale, devi anche concedere le autorizzazioni necessarie per collegare o scollegare tali policy al ruolo IAM associato al nodo principale. In particolare, dovrai allegare le autorizzazioni «iam:AttachRolePolicy» e «iam:DetachRolePolicy» (o il loro equivalente in una policy gestita) al ruolo IAM utilizzato dal nodo principale. Per ulteriori informazioni, consulta [AWS ParallelCluster esempi di politiche utente per la gestione delle risorse IAM](#).

Per ulteriori informazioni sulle politiche predefinite, consulta [AWS Identity and Access Management autorizzazioni in AWS ParallelCluster](#).

#### TemplateUrl (obbligatorio):

L'URL del AWS ParallelCluster file di risorse personalizzato.

## Uscite:

### ServiceToken:

Un valore che può essere utilizzato come ServiceToken proprietà di risorsa personalizzata. Una risorsa personalizzata ServiceToken specifica dove CloudFormation invia le richieste. Si tratta di un input obbligatorio per una risorsa cluster inclusa nel CloudFormation modello.

### LogGroupArn:

L'ARN a CloudWatch LogGroup cui si collega la risorsa sottostante.

### LambdaLayerArn:

L'ARN del livello Lambda utilizzato per eseguire le operazioni. AWS ParallelCluster

## Risorsa del cluster

La risorsa del CloudFormation cluster è formattata come illustrato nel seguente frammento di CloudFormation modello:

```
PclusterCluster:
 Type: Custom::PclusterCluster
 Properties:
 ServiceToken: !GetAtt [PclusterClusterProvider , Outputs.ServiceToken]
 ClusterName: !Sub 'c-${AWS::StackName}' # Must be different from StackName
 ClusterConfiguration:
 # Your Cluster Configuration
```

## Proprietà:

### ServiceToken:

L'output dello stack del AWS ParallelCluster provider. ServiceToken

### ClusterName:

Il nome del cluster da creare e gestire. Il nome non deve corrispondere al nome CloudFormation dello stack. Il nome non può essere modificato dopo la creazione del cluster.

### ClusterConfiguration:

Il file YAML di configurazione del cluster, come descritto in. [File di configurazione del cluster](#) Tuttavia, è possibile utilizzare i CloudFormation costrutti usuali, come le funzioni intrinseche.

### DeletionPolicy:

Definisce se eliminare il cluster quando viene eliminato lo stack principale. Il valore predefinito è `Delete`.

Conserva:

Conserva il cluster se la risorsa personalizzata viene eliminata.

#### Note

Per mantenere funzionante il cluster mantenuto, le risorse dipendenti dal cluster, come lo storage e la rete, devono disporre di una politica di eliminazione impostata da conservare.

Elimina:

Elimina il cluster se la risorsa personalizzata viene eliminata.

### **Fn::GetAtt**valori restituiti:

La funzione `Fn::GetAtt` intrinseca restituisce un valore per un attributo specificato di un tipo. Per ulteriori informazioni sull'uso della `Fn::GetAtt` intrinseca funzione, vedere [Fn::GetAtt](#).

`ClusterProperties`:

I valori dell'[pcluster describe-cluster](#) operazione.

Messaggi di convalida:

Una stringa contenente tutti i messaggi di convalida che si sono verificati durante l'ultima operazione di creazione o aggiornamento.

`logGroupName`:

Il nome del gruppo di log utilizzato per registrare le operazioni del cluster Lambda. Gli eventi di registro vengono conservati per 90 giorni e il gruppo di log viene conservato dopo l'eliminazione del cluster.

### Esempio: `Fn::GetAtt`

```
Provide the public IP address of the head node as an output of a stack
Outputs:
 HeadNodeIp:
 Description: The public IP address of the head node
 Value: !GetAtt [PclusterCluster, headNode.publicIpAddress]
```

Esempio: CloudFormation modello semplice e completo con una risorsa AWS ParallelCluster personalizzata:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
 HeadNodeSubnet:
 Description: Subnet where the HeadNode will run
 Type: AWS::EC2::Subnet::Id

 ComputeSubnet:
 Description: Subnet where the Compute Nodes will run
 Type: AWS::EC2::Subnet::Id

 KeyName:
 Description: KeyPair to login to the head node
 Type: AWS::EC2::KeyPair::KeyName

Resources:
 PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 TemplateURL: !Sub
 - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
 ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
 - { Version: 3.15.0 }

 PclusterCluster:
 Type: Custom::PclusterCluster
 Properties:
 ServiceToken: !GetAtt [PclusterClusterProvider , Outputs.ServiceToken]
 ClusterName: !Sub 'c-${AWS::StackName}'
 ClusterConfiguration:
 Image:
 Os: alinux2
 HeadNode:
```

```

 InstanceType: t2.medium
 Networking:
 SubnetId: !Ref HeadNodeSubnet
 Ssh:
 KeyName: !Ref KeyName
 Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-cr0
 InstanceType: t2.micro
 Networking:
 SubnetIds:
 - !Ref ComputeSubnet

```

#### Outputs:

```

HeadNodeIp:
 Description: The Public IP address of the HeadNode
 Value: !GetAtt [PclusterCluster, headNode.publicIpAddress]
ValidationMessages:
 Description: Any warnings from cluster create or update operations.
 Value: !GetAtt PclusterCluster.validationMessages

```

Per ulteriori informazioni su come utilizzare la risorsa CloudFormation AWS ParallelCluster personalizzata, consulta [Creazione di un cluster con CloudFormation](#).

## Operazioni sui cluster

Quando una risorsa personalizzata del cluster viene aggiunta a uno CloudFormation stack, CloudFormation può eseguire le seguenti operazioni di cluster:

- CloudFormation crea un cluster in un nuovo stack separato quando distribuisce uno stack che include la risorsa personalizzata. AWS ParallelCluster
- Se aggiorni la configurazione del cluster definita nello stack, in base alle politiche di aggiornamento della configurazione, aggiorna il cluster. CloudFormation Il provider di risorse AWS ParallelCluster personalizzate non interrompe la flotta di elaborazione prima di aggiornare il cluster. Ti consigliamo di utilizzare l'[QueueUpdateStrategy](#) impostazione per gli aggiornamenti del cluster. In questo modo, è possibile evitare di effettuare `pcluster update-compute-fleet` chiamate esplicite prima e dopo gli aggiornamenti quando si utilizza la risorsa AWS ParallelCluster personalizzata.
- Se si elimina lo stack, il cluster viene eliminato.

## Risoluzione dei problemi relativi agli stack che includono la risorsa personalizzata AWS ParallelCluster

Con una risorsa AWS ParallelCluster personalizzata, CloudFormation distribuisce un cluster da un nuovo stack separato. È possibile monitorare la creazione del cluster eseguendo le seguenti operazioni:

1. Vai CloudFormation a Console di gestione AWS e scegli Stacks nel riquadro di navigazione.
2. Scegli lo stack con il nome che hai definito per il nome del cluster.
3. Se lo stato dello stack è `ROLLBACK_COMPLETE`, si è verificato un errore durante la creazione del cluster.
4. Scegli i dettagli dello stack e scegli la scheda Eventi.
5. Cerca in Events on Logical ID il nome che hai definito per il nome del cluster. Ha un `Status reason` che indica il motivo del problema.
6. Puoi anche scegliere il menu a discesa Pile, quindi Elimina per visualizzare l'elenco degli stack eliminati. Seleziona lo stack con il nome del cluster e visualizza gli eventi per maggiori dettagli.
7. Per visualizzare l'output del provider di risorse personalizzato che gestisce il cluster, seleziona lo stack con la descrizione "Risorsa personalizzata del AWS ParallelCluster cluster». Scegli la scheda Risorse, trova la risorsa con ID **PclusterCfnFunctionLogGroup** logico e segui il link fornito. Visualizza i flussi di log che mostrano l'output di debug Lambda.
8. Per risolvere i problemi relativi al cluster, consulta [AWS ParallelCluster risoluzione dei problemi](#)

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) è un dispositivo di rete che dispone di funzionalità di bypass del sistema operativo per comunicazioni di rete a bassa latenza con altre istanze sulla stessa sottorete. EFA è esposto utilizzando Libfabric e può essere utilizzato dalle applicazioni che utilizzano l'interfaccia MPI (Messaging Passing Interface).

Per utilizzare EFA con AWS ParallelCluster e uno Slurm scheduler, imposta [SlurmQueues//ComputeResourcesEfa/Enabled](#) su `true`

Per visualizzare l'elenco delle istanze Amazon EC2 che supportano EFA, consulta [Tipi di istanze supportati nella Guida per l'utente di Amazon EC2 per le istanze Linux](#).

Ti consigliamo di eseguire le istanze abilitate per EFA in un gruppo di collocamento. In questo modo le istanze vengono lanciate in un gruppo a bassa latenza in un'unica zona di disponibilità. Per ulteriori informazioni su come configurare i gruppi di collocamento con AWS ParallelCluster, vedere [SlurmQueues](#)//. [NetworkingPlacementGroup](#)

#### Note

Elastic Fabric Adapter (EFA) non è supportato in diverse zone di disponibilità. Per ulteriori informazioni, consulta [Scheduling](#)//[Networking SlurmQueues](#)/. [SubnetIds](#)

#### Note

Per impostazione predefinita, Ubuntu le distribuzioni abilitano la protezione ptrace (traccia del processo). ptrace la protezione è disabilitata in modo che Libfabric funzioni correttamente. Per ulteriori informazioni, consulta [Disabilita la protezione ptrace](#) nella Guida per l'utente di Amazon EC2.

## Configurazione di rete EFA predefinita

A partire dalla versione AWS ParallelCluster 3.15.0, quando EFA è abilitato, configura AWS ParallelCluster automaticamente le interfacce di rete solo EFA per separare il traffico EFA dal traffico IP. Ciò massimizza la larghezza di banda EFA riducendo al minimo il consumo di indirizzi IP. AWS ParallelCluster determina la configurazione ottimale in base alle funzionalità del tipo di istanza.

Questa configurazione predefinita è consigliata per la maggior parte dei carichi di lavoro, tra cui HPC strettamente abbinato e formazione distribuita. AI/ML

## Personalizzazione delle interfacce di rete EFA

Se il carico di lavoro richiede una configurazione di rete diversa, ad esempio l'ottimizzazione della larghezza di banda EFA sulle schede di rete secondarie o la configurazione di un sottoinsieme di schede di rete disponibili, è possibile sovrascrivere le impostazioni predefinite utilizzando il parametro//. [SlurmQueuesComputeResourcesLaunchTemplateOverrides](#) Ciò sostituisce l'intera configurazione dell'interfaccia di rete dei nodi di calcolo con la configurazione definita nel modello di avvio.



Per una step-by-step procedura dettagliata, consulta [the section called “Personalizza le interfacce di rete dei nodi di calcolo con sostituzioni dei modelli di avvio”](#)

#### Warning

Se configuri le interfacce di rete in un modo non supportato dal tipo di istanza, le istanze non verranno avviate. Per verificare le configurazioni di rete supportate per il tuo tipo di istanza, consulta [DescribeInstanceTypes](#) il riferimento alle API di Amazon EC2.

Per ulteriori informazioni, consulta [Elastic Fabric Adapter](#) nella Guida per l'utente di Amazon EC2 e [ridimensiona i carichi di lavoro HPC con Elastic Fabric Adapter e AWS ParallelCluster](#) nel blog Open Source.AWS

## Abilitare Intel MPI

Intel MPI è disponibile su. AWS ParallelCluster AMIs

#### Note

Per utilizzare Intel MPI, è necessario riconoscere e accettare i termini della licenza software [semplificata Intel](#).

Per impostazione predefinita, Open MPI viene inserito nel percorso. Per abilitare Intel MPI anziché Open MPI, è necessario prima caricare il modulo Intel MPI. Quindi, è necessario installare la versione più recente utilizzando `module load intelmpi` Il nome esatto del modulo viene modificato con ogni aggiornamento. Per vedere quali moduli sono disponibili, eseguire `module avail`. L'output è il seguente.

```
$ module avail
-----/usr/share/Modules/modulefiles

dot modules
libfabric-aws/1.16.0~amzn3.0 null
module-git openmpi/4.1.4
module-info use.own
```

```
-----/opt/intel/mpi/2021.6.0/modulefiles

intelmpi
```

Per caricare un modulo, eseguire `module load modulename`. È possibile aggiungere questo allo script utilizzato per l'esecuzione di `mpirun`.

```
$ module load intelmpi
```

Per vedere quali moduli sono caricati, eseguire `module list`.

```
$ module list
Currently Loaded Modulefiles:
 1) intelmpi
```

Per verificare che Intel MPI sia abilitato, eseguire `mpirun --version`.

```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2021.6 Build 20220227 (id: 28877f3f32)
Copyright 2003-2022, Intel Corporation.
```

Dopo il caricamento del modulo Intel MPI, vengono modificati più percorsi per utilizzare gli strumenti Intel MPI. Per eseguire il codice che è stato compilato dagli strumenti Intel MPI, per prima cosa caricare il modulo MPI.

#### Note

Intel MPI non è compatibile con le istanze basate su AWS Graviton.

#### Note

Prima della AWS ParallelCluster versione 2.5.0, Intel MPI non era disponibile AWS ParallelCluster AMIs nelle regioni Cina (Pechino) e Cina (Ningxia).

## AWS ParallelCluster API

Che cos'è l'API AWS ParallelCluster ?

AWS ParallelCluster L'API è un'applicazione serverless che, una volta implementata Account AWS, fornisce l'accesso programmatico alle AWS ParallelCluster funzionalità tramite un'API.

AWS ParallelCluster L'API è distribuita come [CloudFormation](#) modello autonomo che include un endpoint [Amazon API Gateway](#), che espone le AWS ParallelCluster funzionalità, e una [AWS Lambda](#) funzione che si occupa dell'elaborazione delle funzionalità richiamate.

L'immagine seguente mostra un diagramma di architettura di alto livello dell'infrastruttura API. AWS ParallelCluster

## AWS ParallelCluster Documentazione API

Il file delle specifiche OpenAPI che descrive l' AWS ParallelCluster API può essere scaricato da:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

A partire dal file delle specifiche OpenAPI, puoi utilizzare uno dei tanti strumenti disponibili come [Swagger UI](#) o [Redoc](#) per generare documentazione per l'API. AWS ParallelCluster

Come implementare l'API AWS ParallelCluster

Per implementare AWS ParallelCluster l'API devi essere un amministratore di Account AWS

Il modello utilizzato per distribuire l'API è disponibile al seguente URL:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

dove **<REGION>** è il Regione AWS luogo in cui deve essere distribuita l'API ed **<VERSION>** è la AWS ParallelCluster versione (ad esempio 3.15.0).

AWS Lambda utilizza un'interfaccia di livello Lambda con [AWS ParallelCluster API della libreria Python](#) per elaborare le funzionalità richiamate dall'API.

**⚠ Warning**

Qualsiasi utente in Account AWS, che dispone di accesso privilegiato ai AWS Lambda servizi Amazon API Gateway, eredita automaticamente le autorizzazioni per AWS ParallelCluster amministrare le risorse API.

## Implementa l'API con AWS ParallelCluster AWS CLI

In questa sezione, imparerai come eseguire la distribuzione con AWS CLI

Configura AWS le credenziali da utilizzare con la CLI se non l'hai già fatto.

```
$ aws configure
```

Esegui i seguenti comandi per distribuire l'API:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This can be any name
$ VERSION=3.15.0
$ aws cloudformation create-stack \
 --region ${REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-create-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

Personalizza la tua implementazione

Puoi utilizzare i CloudFormation parametri esposti dal modello per personalizzare la distribuzione dell'API. Per configurare il valore di un parametro durante la distribuzione tramite la CLI, è possibile utilizzare la seguente opzione: `--parameters ParameterKey=KeyName,ParameterValue=Value`

I parametri seguenti sono facoltativi:

- **Regione:** utilizza il Region parametro per specificare se l'API è in grado di controllare tutte le risorse Regioni AWS (impostazione predefinita) o singolarmente. Regione AWS Imposta questo valore Regione AWS sull'API su cui viene distribuita per limitare l'accesso.

- **ParallelClusterFunctionRole**- Questo sostituisce il ruolo IAM assegnato alla AWS Lambda funzione che implementa le funzionalità. AWS ParallelCluster Il parametro accetta l'ARN di un ruolo IAM. Questo ruolo deve essere configurato per essere utilizzato AWS Lambda come principale IAM. Inoltre, poiché questo ruolo sostituirà il ruolo predefinito della funzione API Lambda, deve disporre almeno delle autorizzazioni predefinite richieste dall'API come elencato in. [AWS ParallelCluster esempi di politiche pcluster utente](#)
- **ParallelClusterFunctionAdditionalPolicies**- ARN della policy IAM aggiuntiva da allegare al ruolo della funzione AWS ParallelCluster API. È possibile specificare una sola politica.
- **CustomDomainName**, **CustomDomainCertificate**, **CustomDomainHostedZoneId** - Utilizza questi parametri per impostare un dominio personalizzato per l'endpoint Amazon API Gateway. **CustomDomainName** è il nome del dominio da utilizzare, **CustomDomainCertificate** è l'ARN di un certificato AWS gestito per questo nome di dominio ed **CustomDomainHostedZoneId** è l'ID della zona ospitata di [Amazon Route 53](#) in cui desideri creare record.

 Warning

Puoi configurare impostazioni di dominio personalizzate per applicare una versione minima di Transport Layer Security (TLS) per l'API. Per ulteriori informazioni, consulta [Scelta di una versione TLS minima per un dominio personalizzato in API Gateway](#).

- **EnableIamAdminAccess**- Per impostazione predefinita, la AWS Lambda funzione che elabora le operazioni AWS ParallelCluster API è configurata con un ruolo IAM che impedisce qualsiasi accesso IAM privilegiato (`EnableIamAdminAccess=false`). Ciò rende l'API incapace di elaborare operazioni che richiedono la creazione di ruoli o politiche IAM. Per questo motivo, la creazione di cluster o immagini personalizzate ha esito positivo solo quando i ruoli IAM vengono forniti come input come parte della configurazione delle risorse.

Quando `EnableIamAdminAccess` è impostata `true` sull' AWS ParallelCluster API, vengono concesse le autorizzazioni per gestire la creazione dei ruoli IAM necessari per distribuire cluster o generare contenuti personalizzati. AMIs

 Warning

Quando è impostato su `true`, concede i privilegi di amministratore IAM alla funzione che elabora le AWS Lambda operazioni. AWS ParallelCluster

Fai riferimento a [AWS ParallelCluster esempi di politiche utente per la gestione delle risorse IAM](#) per ulteriori dettagli sulle funzionalità che possono essere sbloccate quando abiliti questa modalità.

- **PermissionsBoundaryPolicy**- Questo parametro opzionale accetta un ARN della policy IAM esistente che verrà impostato come limite di autorizzazioni per tutti i ruoli IAM creati dall'infrastruttura API PC e come condizione per le autorizzazioni amministrative IAM in modo che solo i ruoli con questa policy possano essere creati dall'API PC.

[Modalità PermissionsBoundary](#) Per ulteriori dettagli sulle restrizioni imposte da questa modalità, fare riferimento a.

- **CreateApiUserRole**- Per impostazione predefinita, l'implementazione dell' AWS ParallelCluster API include la creazione di un ruolo IAM impostato come unico ruolo autorizzato a richiamare l'API. L'endpoint Amazon API Gateway è configurato con una policy basata sulle risorse per concedere l'autorizzazione di richiamo solo all'utente creato. Per modificare questa impostazione, imposta `CreateApiUserRole=false` e quindi concedi l'accesso all'API a utenti IAM selezionati. Per ulteriori informazioni, consulta [Controllare l'accesso per richiamare un'API nella Amazon API Gateway Developer Guide](#).

#### Warning

Quando `CreateApiUserRole=true` l'accesso all'endpoint API non è limitato dalle policy delle risorse di Amazon API Gateway, tutti i ruoli IAM che dispongono di `execute-api:Invoke` autorizzazioni non vincolate possono accedere alle funzionalità. AWS ParallelCluster Per ulteriori informazioni, consulta [Controllare l'accesso a un'API con le policy delle risorse di API Gateway](#) nella API Gateway Developer Guide.

#### Warning

`ParallelClusterApiUserRole` È autorizzato a richiamare tutte le operazioni AWS ParallelCluster API. Per limitare l'accesso a un sottoinsieme di risorse API, consulta il [metodo API Control who can call an API Gateway with IAM policies](#) nella API Gateway Developer Guide.

- **IAMRoleAndPolicyPrefix**- Questo parametro opzionale accetta una stringa contenente un massimo di 10 caratteri che verrà utilizzata come prefisso sia per i ruoli IAM che per le policy create come parte dell'infrastruttura API PC.

## Aggiornamento dell'API

In questa sezione, imparerai come utilizzare una delle due opzioni disponibili per aggiornare l'API.

Aggiornamento a una versione più recente AWS ParallelCluster

Opzione 1: per rimuovere l'API esistente, elimina lo CloudFormation stack corrispondente e distribuisce la nuova API come mostrato sopra.

Opzione 2: per aggiornare l'API esistente, esegui i seguenti comandi:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.15.0
$ aws cloudformation update-stack \
 --region ${REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-update-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

## Invocare l'API AWS ParallelCluster

L'endpoint AWS ParallelCluster Amazon API Gateway è configurato con un [tipo di AWS IAM autorizzazione](#) e richiede che tutte le richieste siano firmate SigV4 con credenziali IAM valide ([riferimento API: creazione](#) di richieste http).

Se distribuito con impostazioni predefinite, le autorizzazioni di richiamo dell'API vengono concesse solo all'utente IAM predefinito creato con l'API.

Per recuperare l'ARN dell'utente IAM predefinito, esegui:

```
$ REGION=<region>
```

```
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiUserRole'].OutputValue" --
output text
```

[Per ottenere credenziali temporanee per l'utente IAM predefinito, esegui il comando STS.AssumeRole](#)

Per recuperare l'endpoint dell' AWS ParallelCluster API esegui il seguente comando:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiInvokeUrl'].OutputValue" --
output text
```

L' AWS ParallelCluster API può essere richiamata da qualsiasi client HTTP conforme alle specifiche OpenAPI disponibili qui:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

[Le richieste devono essere firmate con SigV4 come documentato qui.](#)

Al momento, non offriamo alcuna implementazione ufficiale del client API. Tuttavia, puoi utilizzare [OpenAPI Generator](#) per generare facilmente client API dal modello OpenAPI. Una volta generato il client, è necessario aggiungere la firma SigV4, se non fornita immediatamente.

[Un'implementazione di riferimento per un client API Python è disponibile nel AWS ParallelCluster repository.](#) Per saperne di più su come utilizzare il client API Python, consulta il [Utilizzo dell' AWS ParallelCluster API](#) tutorial.

[Per implementare meccanismi di controllo degli accessi più avanzati, come Amazon Cognito o Lambda Authorizers, o per proteggere ulteriormente l'API con le nostre chiavi API, segui la AWS WAF documentazione di Amazon API Gateway.](#)

#### Warning

Un utente IAM autorizzato a richiamare l' AWS ParallelCluster API può controllare indirettamente tutte le AWS risorse gestite da in. AWS ParallelCluster Account AWS



Ciò include la creazione di AWS risorse che l'utente non può controllare direttamente a causa delle restrizioni sulla politica IAM dell'utente. Ad esempio, la creazione di un AWS ParallelCluster cluster, a seconda della sua configurazione, potrebbe includere la distribuzione di istanze Amazon EC2, Amazon Route 53, file system Amazon Elastic File System, file system Amazon FSx , ruoli IAM e risorse di altri Servizi AWS utenti su AWS ParallelCluster cui l'utente potrebbe non avere il controllo diretto.

### Warning

Quando crei un cluster con una configurazione `AdditionalIamPolicies` specificata, le politiche aggiuntive devono corrispondere a uno dei seguenti modelli:

```
- !Sub arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/parallelcluster*
- !Sub arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/parallelcluster/*
- !Sub arn:${AWS::Partition}:iam::aws:policy/CloudWatchAgentServerPolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonSSMManagedInstanceCore
- !Sub arn:${AWS::Partition}:iam::aws:policy/AWSBatchFullAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonS3ReadOnlyAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/AWSBatchServiceRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/EC2InstanceProfileForImageBuilder
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole
```

Se sono necessarie altre politiche aggiuntive, è possibile effettuare una delle seguenti operazioni:

- Modifica il file `DefaultParallelClusterIamAdminPolicy` in:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

Aggiungi la politica nella `ArnLike/iam:PolicyARN` sezione.

- Non specificate le policy per `AdditionalIamPolicies` nel file di configurazione e aggiungete manualmente le policy all' AWS ParallelCluster Instance Role creato all'interno del cluster.

## Accesso ai log e alle metriche dell'API

I log delle API vengono pubblicati su Amazon CloudWatch con una conservazione di 30 giorni. Per recuperare il LogGroup nome associato a una distribuzione API, esegui il seguente comando:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --
 stack-name ${API_STACK_NAME} --query "Stacks[0].Outputs[?
 OutputKey=='ParallelClusterLambdaLogGroup'].OutputValue" --output text
```

È possibile accedere alle metriche, ai log e ai log di [AWS X-Ray](#) traccia Lambda anche tramite la console Lambda. Per recuperare l'ARN della funzione Lambda associata a una distribuzione API, esegui il comando seguente:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterLambdaArn'].OutputValue" --
 output text
```

## AWS ParallelCluster per Terraform

[A partire dalla AWS ParallelCluster 3.8.0, puoi distribuire cluster e immagini personalizzate utilizzando Terraform.](#) Per iniziare a utilizzare questa funzionalità, consulta [Terraform Provider for dal registro Terraform](#). AWS ParallelCluster

### Note

È necessario che [ParallelCluster l'API](#) sia implementata nel proprio account per utilizzare il provider.

Utilizza la tabella seguente per determinare la compatibilità tra il provider e le AWS ParallelCluster versioni:

| Versione del provider | AWS ParallelCluster versione |
|-----------------------|------------------------------|
| 1.0.0                 | 3.8.0-3.10.1                 |
| 1.1.0                 | 3,11,0+                      |

Vedi [esempi](#) di come utilizzare il provider.

Per un'esperienza ancora più fluida, usa il [modulo ufficiale Terraform di Terraform AWS ParallelCluster](#) Registry. Il modulo consente di implementare:

1. ParallelCluster API
2. ParallelCluster cluster definiti con file di configurazione YAML e HCL
3. infrastruttura di rete richiesta da un cluster ParallelCluster

Guarda [alcuni esempi](#) di utilizzo del modulo.

## Connect ai nodi principali e di accesso tramite Amazon DCV

Amazon DCV è una tecnologia di visualizzazione remota che consente agli utenti di connettersi in modo sicuro ad applicazioni 3D a uso intensivo di grafica ospitate su un server remoto ad alte prestazioni. Per ulteriori informazioni, consulta [Amazon DCV](#).

[Il software Amazon DCV viene installato automaticamente sul nodo principale e può essere abilitato utilizzando la Dcv sezione di. HeadNode](#)

[A partire dalla AWS ParallelCluster versione 3.11, Amazon DCV può essere abilitato per un pool di nodi di accesso utilizzando la sezione Dcv dalla configurazione Sezione LoginNodes/Pools.](#)

```
LoginNodes:
 Pools:
 Dcv:
 Enabled: true
```

AWS ParallelCluster imposta `/home/<DEFAULT_AMI_USER>` come cartella di archiviazione del server [DCV](#). Per ulteriori informazioni sui parametri di configurazione di Amazon DCV, consulta [HeadNode/Dcv](#).

Per connetterti alla sessione Amazon DCV sul nodo principale, usa il [dcv-connect](#) comando. Per connetterti a un nodo di accesso, utilizza `dcv-connect` il `--login-node-ip` parametro e inserisci l'indirizzo IP pubblico o privato del nodo di accesso a cui desideri connetterti.

## Certificato HTTPS Amazon DCV

Amazon DCV genera automaticamente un certificato autofirmato per proteggere il traffico tra il client Amazon DCV e il server Amazon DCV.

Per sostituire il certificato Amazon DCV autofirmato predefinito con un altro certificato, connettiti prima al nodo principale. Quindi, copiare il certificato e la chiave nella cartella `/etc/dcv` prima di eseguire il comando [pcluster dcv-connect](#).

Per ulteriori informazioni, consulta [Modifica del certificato TLS](#) nella Amazon DCV Administrator Guide.

## Licenze Amazon DCV

Il server Amazon DCV non richiede un server di licenza quando viene eseguito su EC2 istanze Amazon. Tuttavia, il server Amazon DCV deve connettersi periodicamente a un bucket Amazon S3 per determinare se è disponibile una licenza valida.

AWS ParallelCluster aggiunge automaticamente le autorizzazioni richieste alla policy IAM del nodo principale. Quando utilizzi una IAM Instance Policy personalizzata, utilizza le autorizzazioni descritte in [Amazon DCV on Amazon EC2 nella Amazon](#) DCV Administrator Guide.

Per suggerimenti sulla risoluzione dei problemi, consulta [Risoluzione dei problemi in Amazon DCV](#).

## Uso di **pcluster update-cluster**

In AWS ParallelCluster 3.x, [pcluster update-cluster](#) analizza le impostazioni utilizzate per creare il cluster corrente e le impostazioni nel file di configurazione per individuare eventuali problemi. Se vengono rilevati problemi, vengono segnalati e vengono visualizzati i passaggi da eseguire per risolverli. Ad esempio, se l'elaborazione [InstanceType](#) viene modificata, è necessario interrompere la flotta di elaborazione prima di procedere con l'aggiornamento. Questo problema viene segnalato

quando viene scoperto. Se non vengono rilevati problemi di blocco, viene avviato il processo di aggiornamento e le modifiche vengono segnalate.

È possibile utilizzarli [pcluster update-cluster --dryrun](#) option per visualizzare le modifiche prima della loro esecuzione. Per ulteriori informazioni, consulta [Esempi di pcluster update-cluster](#).

Per una guida alla risoluzione dei problemi, vedere [AWS ParallelCluster risoluzione dei problemi](#).

## Politica di aggiornamento: definizioni

Politica di aggiornamento: i nodi di accesso nel cluster devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

Non è possibile modificare queste impostazioni mentre i nodi di accesso nel cluster sono in uso. È necessario annullare la modifica oppure interrompere i nodi di accesso del cluster. (È possibile interrompere i nodi di accesso nel cluster impostando il numero di ciascun pool pari a 0). Dopo aver interrotto i nodi di accesso del cluster, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

### Note

Questa politica di aggiornamento è supportata a partire dalla versione 3.7.0 AWS ParallelCluster .

Politica di aggiornamento: è possibile aggiungere pool di nodi di accesso, ma la rimozione di un pool richiede l'interruzione di tutti i nodi di accesso nel cluster.

Per rimuovere un pool, è necessario arrestare tutti i nodi di accesso nel cluster. (È possibile interrompere i nodi di accesso nel cluster impostando il conteggio di ogni pool pari a 0). Dopo che i nodi di accesso del cluster sono stati interrotti, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

### Note

Questa politica di aggiornamento è supportata a partire dalla AWS ParallelCluster versione 3.11.0.

Politica di aggiornamento: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

Non è possibile modificare queste impostazioni mentre i nodi di accesso nel pool sono in uso. È necessario annullare la modifica o interrompere i nodi di accesso del pool. (È possibile interrompere i nodi di accesso nel pool impostando il conteggio del pool pari a 0). Dopo che i nodi di accesso del pool sono stati interrotti, è possibile aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

 Note

Questa politica di aggiornamento è supportata a partire dalla AWS ParallelCluster versione 3.11.0.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Dopo aver modificato questa impostazione, il cluster può essere aggiornato utilizzando [pcluster update-cluster](#).

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

Dopo aver modificato questa impostazione, il cluster non può essere aggiornato. È necessario ripristinare le impostazioni per il cluster originale e creare un nuovo cluster con le impostazioni aggiornate. È possibile eliminare il cluster originale in un secondo momento. Per creare il nuovo cluster, usa [pcluster create-cluster](#). Per eliminare il cluster originale, utilizzare [pcluster delete-cluster](#).

Politica di aggiornamento: questa impostazione non viene analizzata durante un aggiornamento.

Queste impostazioni possono essere modificate e il cluster può essere aggiornato utilizzando [pcluster update-cluster](#).

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

Queste impostazioni non possono essere modificate mentre esiste la flotta di elaborazione. La modifica deve essere annullata o la flotta di elaborazione deve essere interrotta (in uso). [pcluster update-compute-fleet](#) Dopo l'arresto del parco di elaborazione, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche. Ad esempio, se si utilizza uno

Slurm scheduler con [SlurmQueues/- ComputeResourcesName/MinCount](#)> 0, viene avviato un parco di elaborazione.

Politica di aggiornamento: la flotta di elaborazione e i nodi di accesso devono essere interrotti affinché questa impostazione venga modificata per un aggiornamento.

Queste impostazioni non possono essere modificate mentre esiste il parco di elaborazione o se i nodi di accesso sono in uso. La modifica deve essere annullata oppure il parco di elaborazione e i nodi di accesso devono essere interrotti (è possibile interrompere l'utilizzo del parco di elaborazione). [pcluster update-compute-fleet](#) Dopo aver interrotto il parco di elaborazione e i nodi di accesso, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

Politica di aggiornamento: questa impostazione non può essere ridotta durante un aggiornamento.

Queste impostazioni possono essere modificate, ma non possono essere ridotte. Se è necessario ridurre queste impostazioni, è necessario ripristinare le impostazioni per il cluster originale e creare un nuovo cluster con le impostazioni aggiornate. È possibile eliminare il cluster originale in un secondo momento. Per creare il nuovo cluster, usa [pcluster create-cluster](#). Per eliminare il cluster originale, utilizzare [pcluster delete-cluster](#).

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito. Se si forza l'aggiornamento, il nuovo valore verrà ignorato e verrà utilizzato il vecchio valore.

Dopo aver modificato questa impostazione, il cluster non può essere aggiornato. È necessario ripristinare le impostazioni per il cluster originale e creare un nuovo cluster con le impostazioni aggiornate. È possibile eliminare il cluster originale in un secondo momento. Per creare il nuovo cluster, usa [pcluster create-cluster](#). Per eliminare il cluster originale, utilizzare [pcluster delete-cluster](#).

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o [QueueUpdateStrategy](#) deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

Queste impostazioni possono essere modificate. La flotta di elaborazione deve essere interrotta (in uso [pcluster update-compute-fleet](#)) o [QueueUpdateStrategy](#) deve essere impostata. Dopo l'arresto o [QueueUpdateStrategy](#) l'impostazione del parco di elaborazione, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

 Note

Questa politica di aggiornamento è supportata a partire dalla AWS ParallelCluster versione 3.2.0.

Politica di aggiornamento: per questa impostazione dei valori dell'elenco, è possibile aggiungere un nuovo valore durante un aggiornamento oppure è necessario interrompere il parco di elaborazione quando si rimuove un valore esistente.

Un nuovo valore per queste impostazioni può essere aggiunto durante un aggiornamento. Dopo aver aggiunto un nuovo valore all'elenco, il cluster può essere aggiornato utilizzando ([pcluster update-cluster](#)).

Per rimuovere un valore esistente dall'elenco, è necessario interrompere (utilizzando [pcluster update-compute-fleet](#)) il parco di calcolo.

Ad esempio, se utilizzi uno Slurm scheduler e aggiungi un nuovo tipo di istanza a [Instances/ InstanceType](#), puoi aggiornare il cluster senza interrompere il parco di calcolo. [Per rimuovere un tipo di istanza esistente da Instances/ InstanceType, è necessario prima arrestare il parco di calcolo \(utilizzando pcluster\). update-compute-fleet](#)

 Note

Questa politica di aggiornamento è supportata a partire dalla versione 3.2.0. AWS ParallelCluster

Politica di aggiornamento: la riduzione delle dimensioni di una coda richiede l'arresto della flotta di elaborazione o [QueueUpdateStrategy](#) deve essere impostata su TERMINATE affinché questa impostazione venga modificata per un aggiornamento.

Queste impostazioni possono essere modificate, ma se la modifica ridurrebbe le dimensioni della coda, la flotta di elaborazione deve essere interrotta (usando [pcluster update-compute-fleet](#)) o [QueueUpdateStrategy](#) deve essere impostata su TERMINATE. Dopo che la flotta di elaborazione è stata interrotta o [QueueUpdateStrategy](#) è stata impostata su TERMINATE, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.



Il comando `TERMINATE` impostato durante il ridimensionamento della capacità del cluster interromperà solo i nodi dalla parte posteriore dell'elenco dei nodi e lascerà intatti tutti gli altri nodi della stessa partizione.

Ad esempio, se la capacità iniziale del cluster è `e`, i nodi sono `st-[1-5]; dy-[1-5]`. Quando si ridimensiona il cluster in `MinCount = 3` e `MaxCount = 5`, la nuova capacità del cluster sarà composta dai nodi `st-[1-3]; dy-[1-2]`, che non verranno toccati durante l'aggiornamento. Solo i nodi `st-[4-5]; dy-[3-5]` verranno terminati durante l'aggiornamento.

Le seguenti modifiche sono supportate e non richiedono l'arresto del parco di elaborazione né l'[QueueUpdateStrategy](#) impostazione su `TERMINATE`:

- Ne è stata aggiunta una nuova [SlurmQueue](#)
- [ComputeResource](#) Viene aggiunto un nuovo
- [MaxCount](#) è aumentato
- [MinCount](#) è aumentato ed [MaxCount](#) è aumentato almeno dello stesso importo

Nota: questa politica di aggiornamento è supportata a partire dalla AWS ParallelCluster versione 3.9.0.

Politica di aggiornamento: per questa impostazione dei valori dell'elenco, la flotta di elaborazione deve essere interrotta o [QueueUpdateStrategy](#) deve essere impostata per aggiungere un nuovo valore; la flotta di elaborazione deve essere interrotta quando si rimuove un valore esistente.

Un nuovo valore per queste impostazioni può essere aggiunto durante un aggiornamento. La flotta di elaborazione deve essere interrotta (in uso [pcluster update-compute-fleet](#)) o [QueueUpdateStrategy](#) deve essere impostata. Dopo l'arresto o [QueueUpdateStrategy](#) l'impostazione del parco di elaborazione, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

Per rimuovere un valore esistente dall'elenco, è necessario interrompere (utilizzare [pcluster update-compute-fleet](#)) il parco di calcolo.

#### Note

Questa politica di aggiornamento è supportata a partire dalla AWS ParallelCluster versione 3.3.0.

Politica di aggiornamento: tutti i nodi di calcolo devono essere interrotti per l'eliminazione di un gruppo di posizionamento gestito. La flotta di elaborazione deve essere interrotta o [QueueUpdateStrategy](#) deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

La flotta di elaborazione deve essere interrotta (utilizzata [pcluster update-compute-fleet](#)) per rimuovere un gruppo di collocamento gestito. Se esegui un aggiornamento del cluster per rimuovere un gruppo di collocamento gestito prima di interrompere il parco di calcolo, viene restituito un messaggio di configurazione non valido e l'aggiornamento non procede. L'arresto del parco di elaborazione garantisce che nessuna istanza sia in esecuzione.

## Esempi di `pcluster update-cluster`

Queste impostazioni possono essere modificate, ma se la modifica ridurrebbe la dimensione della coda, la flotta di elaborazione deve essere interrotta (usando `pcluster update-compute-fleet`) o [QueueUpdateStrategy](#) deve essere impostata su `TERMINATE`. Dopo che la flotta di elaborazione è stata interrotta o [QueueUpdateStrategy](#) è stata impostata su `TERMINATE`, puoi aggiornare il cluster ([pcluster update-cluster](#)) per attivare le modifiche.

- Questo esempio dimostra un aggiornamento con alcune modifiche consentite e l'aggiornamento viene avviato direttamente.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
 "cluster": {
 "clusterName": cluster_name,
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "cloudformationStackArn": stack_arn,
 "region": "us-east-1",
 "version": "3.15.0",
 "clusterStatus": "UPDATE_IN_PROGRESS"
 },
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
```

```

 "sg-0cd61884c4ad16341"
]
}
]
}

```

- Questo esempio dimostra un aggiornamento dryrun con alcune modifiche consentite. Dryrun è utile per segnalare il set di modifiche senza avviare l'aggiornamento.

```

$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1 --dryrun true
{
 "message": "Request would have succeeded, but DryRun flag is set.",
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 }
]
}

```

- Questo esempio dimostra un aggiornamento con alcune modifiche che bloccano l'aggiornamento.

```

$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
 "message": "Update failure",
 "updateValidationErrors": [
 {
 "parameter": "HeadNode.Ssh.KeyName",
 "requestedValue": "mykey_2",
 "message": "Update actions are not currently supported for the 'KeyName'
parameter. Restore 'KeyName' value to 'jenkinsjun'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
 "currentValue": "mykey_1"
 },
 {

```

```

 "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
 "requestedValue": "c4.xlarge",
 "message": "All compute nodes must be stopped. Stop the compute fleet with the
pcluster update-compute-fleet command",
 "currentValue": "t2.micro"
 },
 {
 "parameter": "SharedStorage[ebs1].MountDir",
 "requestedValue": "/my/very/very/long/shared_dir",
 "message": "Update actions are not currently supported for the 'MountDir'
parameter. Restore 'MountDir' value to '/shared'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
 "currentValue": "/shared"
 }
],
"changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 },
 {
 "parameter": "HeadNode.Ssh.KeyName",
 "requestedValue": "mykey_2",
 "currentValue": "mykey_1"
 },
 {
 "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
 "requestedValue": "c4.xlarge",
 "currentValue": "t2.micro"
 },
 {
 "parameter": "SharedStorage[ebs1].MountDir",
 "requestedValue": "/my/very/very/long/shared_dir",
 "currentValue": "/shared"
 }
]

```

```
}
```

## AWS ParallelCluster Personalizzazione AMI

Esistono scenari in cui AWS ParallelCluster è necessario creare un'AMI personalizzata per. Questa sezione illustra gli aspetti da considerare quando si crea un' AWS ParallelCluster AMI personalizzata.

Puoi creare un' AWS ParallelCluster AMI personalizzata utilizzando uno dei seguenti metodi:

1. [Crea un file di configurazione dell'immagine](#) di compilazione, quindi utilizza la `pcluster` CLI per creare l'immagine con EC2 Image Builder. Questo processo è automatizzato, ripetibile e supporta il monitoraggio. Per ulteriori informazioni, consulta i comandi dell'[pcluster](#) immagine.
2. Crea un'istanza da un' AWS ParallelCluster AMI, quindi accedi ad essa e apporta modifiche manuali. Infine, usa Amazon EC2 per creare una nuova AMI dall'istanza modificata. Questo processo richiede meno tempo. Tuttavia, non è automatizzato o ripetibile e non supporta l'uso dei comandi di monitoraggio delle immagini `pcluster` CLI.

Per ulteriori informazioni su questi metodi, consulta. [Creazione di un' AWS ParallelCluster AMI personalizzata](#)

## AWS ParallelCluster Considerazioni sulla personalizzazione delle AMI

Indipendentemente dal modo in cui crei l'immagine personalizzata, ti consigliamo di eseguire test di convalida preliminari e di includere disposizioni per monitorare lo stato dell'immagine in fase di creazione.

Per creare un'AMI personalizzata utilizzando `pcluster`, crei un [file di configurazione dell'immagine di compilazione](#) con una [Image](#) sezione [Build](#) and che [EC2 Image](#) Builder utilizza per creare l'immagine personalizzata. La `Build` sezione specifica di cosa ha bisogno Image Builder per creare l'immagine. Ciò include [ParentImage](#) (immagine di base) e. [Components](#) Un [componente Image Builder](#) definisce una sequenza di passaggi necessari per personalizzare un'istanza prima della creazione di un'immagine o per testare un'istanza lanciata dall'immagine creata. Per ulteriori informazioni, consulta [Creare un componente personalizzato con Image Builder](#) nella Guida per l'utente di EC2 Image Builder.

Quando viene chiamato da `pcluster` [build-image](#) per creare un'immagine personalizzata, Image Builder utilizza la configurazione dell'immagine di compilazione con AWS ParallelCluster il cookbook

per AWS ParallelCluster avviare il tuo. [ParentImage](#) Image Builder scarica i componenti, esegue le fasi di compilazione e convalida, crea l'AMI, avvia un'istanza dall'AMI ed esegue i test. Al termine del processo, Image Builder produce quindi una nuova immagine o un messaggio di interruzione.

## Esegui test di convalida dei componenti personalizzati

Prima di includere un componente Image Builder in una configurazione, testatelo e convalidatelo utilizzando uno dei seguenti metodi. Poiché il processo di Image Builder può richiedere fino a 1 ora, si consiglia di testare prima i componenti. In questo modo è possibile risparmiare una notevole quantità di tempo.

### Script case

Testa lo script in un'istanza in esecuzione, al di fuori del processo di creazione dell'immagine, e verifica che lo script esca con il codice di uscita 0.

### Caso Amazon Resource Name (ARN)

Testa il documento del componente in un'istanza in esecuzione, al di fuori del processo di creazione dell'immagine. Per un elenco dei requisiti, vedere [Gestione componenti](#) nella Guida per l'utente di Image Builder.

Una volta completata la convalida, aggiungi il componente alla configurazione dell'immagine di compilazione

Dopo aver verificato che il componente personalizzato funzioni, aggiungilo al [file di configurazione dell'immagine di compilazione](#).

## Monitora il processo di Image Builder con **pcluster** comandi per facilitare il debug

### [describe-image](#)

Utilizzate questo comando per monitorare lo stato dell'immagine di compilazione.

### [list-image-log-streams](#)

Utilizzate questo comando per ottenere i flussi IDs di log che potete usare per recuperare gli eventi di registro. [get-image-log-events](#)

### [get-image-log-events](#)

Utilizzate questo comando per ottenere il flusso di log degli eventi del processo di compilazione dell'immagine.

Ad esempio, è possibile eseguire la coda degli eventi di creazione delle immagini utilizzando il comando seguente.

```
$ watch -n 1 'pcluster get-image-log-events -i <image-id> \
--log-stream-name/1 <pcluster-version> \
--query "events[*].message" | tail -n 50'
```

### [get-image-stack-events](#)

Utilizzate questo comando per recuperare gli eventi dello stack di immagini per lo stack creato da Image Builder.

### [export-image-logs](#)

Utilizzate questo comando per salvare i registri delle immagini.

Per ulteriori informazioni sui AWS ParallelCluster log e su Amazon CloudWatch, consulta [Amazon CloudWatch Logs crea log di immagini](#) e [CloudWatch Pannello di controllo Amazon](#).

## Altre considerazioni

### Nuove AWS ParallelCluster versioni e personalizzazioni AMIs

Se si crea e si utilizza un'AMI personalizzata, è necessario ripetere i passaggi utilizzati per creare l'AMI personalizzata con ogni nuova AWS ParallelCluster versione.

### Azioni bootstrap personalizzate

Esamina la [Azioni bootstrap personalizzate](#) sezione per determinare se le modifiche che desideri apportare possono essere scritte tramite script e supportate nelle versioni future AWS ParallelCluster .

### Utilizzo di custom AMIs

È possibile specificare custom AMIs nella configurazione del cluster nelle [CustomAmi](#) sezioni [Image/CustomAmi](#) e [SchedulingSlurmQueues](#)// [NameImage](#)//.

Per risolvere i problemi relativi agli avvisi di convalida AMI personalizzati, consulta. [Risoluzione dei problemi relativi alle AMI personalizzate](#)

# Avvio di istanze con On-Demand Capacity Reservations (ODCR)

Con [On-Demand Capacity Reservations \(ODCR\)](#), puoi riservare la capacità per le tue istanze Amazon EC2 in cluster in una zona di disponibilità specifica. In questo modo, puoi creare e gestire le prenotazioni di capacità indipendentemente dagli account di fatturazione offerti da [Savings Plans](#) o dalle [istanze riservate regionali](#).

È possibile configurare il nostro open ODCRtargeted. Open ODCR copre tutte le istanze che corrispondono agli attributi ODCR. Questi attributi sono il tipo di istanza, la piattaforma e la zona di disponibilità. È necessario definire in modo esplicito l'ODCR mirato nella configurazione del cluster. Per determinare se un ODCR è open o targeted, esegui il comando AWS CLI Amazon [describe-capacity-reservation](#)EC2.

Puoi anche creare un ODCR in un gruppo di collocamento del cluster chiamato Cluster [Placement Group On-Demand Capacity Reservation](#) (CPG ODCR).

ODCRs È possibile raggrupparne più di uno in un gruppo di risorse. Questo può essere definito nel file di configurazione del cluster. Per ulteriori informazioni sui gruppi di risorse, consulta [Cosa sono i gruppi di risorse?](#) nella Guida per l'utente di Resource Groups and Tags.

## Utilizzo di ODCR con AWS ParallelCluster

AWS ParallelCluster supporta ODCR aperto. Quando si utilizza un ODCR aperto, non è necessario specificare nulla in. AWS ParallelCluster Le istanze vengono selezionate automaticamente per il cluster. Potete specificare un gruppo di collocamento esistente o farne AWS ParallelCluster creare uno nuovo per voi.

### ODCR nella configurazione del cluster

A partire dalla AWS ParallelCluster versione 3.3.0, puoi definire gli ODCR nel file di configurazione del cluster, senza bisogno di specificare le sostituzioni delle istanze di esecuzione di Amazon EC2.

Inizia creando [prenotazioni di capacità](#) e [gruppi di risorse](#) utilizzando i metodi descritti nella documentazione collegata per ciascuno di essi. È necessario utilizzare i AWS CLI metodi per creare gruppi di prenotazione della capacità. Se si utilizza il Console di gestione AWS, è possibile creare solo gruppi di risorse basati su Tag o Stack. I gruppi di risorse basati su tag e stack non sono supportati da AWS ParallelCluster o al AWS CLI momento del lancio delle istanze con prenotazioni di capacità.



Dopo aver creato le prenotazioni di capacità e i gruppi di risorse, specificali in [SlurmQueues/CapacityReservationTarget](#) o [SlurmQueuesComputeResources/CapacityReservationTarget](#) come mostrato nell'esempio seguente di configurazione del cluster. Sostituisci i valori *values* evidenziati in rosso con i tuoi valori validi.

```
Image:
 Os: os
HeadNode:
 InstanceType: head_node_instance
 Networking:
 SubnetId: public_subnet_id
 Ssh:
 KeyName: key_name
Scheduling:
 Scheduler: scheduler
SlurmQueues:
 - Name: queue1
 Networking:
 SubnetIds:
 - private_subnet_id
ComputeResources:
 - Name: cr1
 Instances:
 - InstanceType: instance
 MaxCount: max_queue_size
 MinCount: max_queue_size
 Efa:
 Enabled: true
 CapacityReservationTarget:
 CapacityReservationResourceGroupArn: capacity_reservation_arn
```

**OBSOLETO/NON CONSIGLIATO:** ODCR mirato con sostituzioni delle istanze Amazon EC2

 Warning

- A partire dalla AWS ParallelCluster versione 3.3.0, non consigliamo questo metodo. Questa sezione funge da riferimento per le implementazioni che utilizzano versioni precedenti.
- Questo metodo non è compatibile con l'allocazione di tipi di istanze multiple con Slurm.

Il supporto per targeted ODCRs è stato aggiunto nella AWS ParallelCluster versione 3.1.1. In questa versione, è stato introdotto un meccanismo che sovrascrive RunInstances i parametri EC2 e trasmette le informazioni sulla prenotazione da utilizzare per ogni risorsa di elaborazione configurata in AWS ParallelCluster. Questo meccanismo è compatibile con ODCR. targeted. Tuttavia, quando si utilizza targeted ODCR, è necessario specificare la configurazione di run-instances override. Targeted ODCRs deve essere definito in modo esplicito nel comando AWS CLI Amazon [run-instances](#)EC2. Per determinare se un ODCR è open o targeted esegui il comando AWS CLI Amazon [describe-capacity-reservation](#)EC2.

ODCRs È possibile raggrupparne più di uno in un gruppo di risorse. Questo può essere usato nell'override delle istanze di esecuzione per indirizzare più istanze contemporaneamente ODCRs .

Se utilizzi un targeted ODCR, puoi specificare un gruppo di collocamento. Tuttavia, è necessario specificare anche una configurazione run-instances sostitutiva.

Supponiamo che abbia AWS creato un targeted ODCR per te o che tu disponga di un set specifico di istanze riservate. Quindi, non potete specificare un gruppo di collocamento. Le regole configurate da AWS potrebbero entrare in conflitto con l'impostazione del gruppo di collocamento. Quindi, se per la vostra candidatura è richiesto un gruppo di collocamento, utilizzate un [CPG ODCR](#). In entrambi i casi, è necessario specificare anche la configurazione di run-instances override.

Se utilizzate un ODCR CPG, dovete specificare la configurazione di run-instances override e dovete specificare lo stesso gruppo di posizionamento nella configurazione del cluster.

## Utilizzo di istanze riservate con AWS ParallelCluster

Le istanze riservate [sono diverse](#) dalle prenotazioni di capacità (ODCR). Esistono [2 tipi di istanze](#) riservate. Un'istanza riservata regionale non riserva capacità. Un'istanza riservata zonale riserva la capacità nella zona di disponibilità specificata.

Se disponi di istanze riservate regionali, non è prevista alcuna prenotazione di capacità e potresti ricevere errori di capacità insufficiente. Se disponi di istanze riservate zonali, hai una riserva di capacità, ma non ci sono parametri run-instances API che puoi utilizzare per specificarle.

Le istanze riservate sono supportate da qualsiasi versione. AWS ParallelCluster Non è necessario specificare nulla in AWS ParallelCluster e le istanze vengono selezionate automaticamente.

Quando si utilizzano istanze riservate zonali, è possibile evitare potenziali errori di capacità insufficiente omettendo la specifica del gruppo di posizionamento nella configurazione del cluster.

## OBSOLETO/NON CONSIGLIATO: utilizzo della **RunInstances** personalizzazione in AWS ParallelCluster 3 per **targeted** On-Demand Capacity Reservations (ODCR)

### Warning

- A partire dalla AWS ParallelCluster versione 3.3.0, questo metodo non è consigliato. Questa sezione funge da riferimento per le implementazioni che utilizzano versioni precedenti.
- Questo metodo non è compatibile con l'allocazione di tipi di istanze multiple con Slurm.

Puoi sovrascrivere i parametri di Amazon RunInstances EC2 per ogni risorsa di calcolo configurata in una coda di cluster. A tale scopo, crea il `/opt/slurm/etc/pcluster/run_instances_overrides.json` file sul nodo principale del cluster con il seguente contenuto di frammenti di codice:

- `${queue_name}` è il nome della coda a cui desiderate applicare gli override.
- `${compute_resource_name}` è la risorsa di calcolo a cui si desidera applicare le sostituzioni.
- `${overrides}` è un oggetto JSON arbitrario che contiene un elenco di RunInstances sostituzioni da utilizzare per la combinazione specifica di coda e tipo di istanza. [La sintassi delle override deve seguire le stesse specifiche documentate in una chiamata run\\_instances boto3.](#)

```
{
 "${queue_name}": {
 "${compute_resource_name}": {
 ${overrides}
 },
 ...
 },
 ...
}
```

Ad esempio, il seguente codice JSON configura il gruppo ODCR da utilizzare per le istanze configurate in `and. group_arn p4d.24xlarge my-queue my-compute-resource`

```
{
 "my-queue": {
 "my-compute-resource": {
```

```

 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationResourceGroupArn": "group_arn"
 }
 }
 }
}

```

Dopo la generazione di questo file JSON, AWS ParallelCluster i demoni responsabili della scalabilità del cluster utilizzano automaticamente la configurazione override per l'avvio delle istanze. Per confermare che i parametri specificati vengano utilizzati per il provisioning, ad esempio, guardate i seguenti file di registro:

- `/var/log/parallelcluster/clustermgtd`(per capacità statica)
- `/var/log/parallelcluster/slurm_resume.log`(per capacità dinamica)

Se i parametri sono corretti, troverai una voce di registro che contiene quanto segue:

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

**OBSOLETO/NON CONSIGLIATO:** crea un cluster con **targeted** On-Demand Capacity Reservations (ODCR)

#### Warning

- A partire dalla AWS ParallelCluster versione 3.3.0, non consigliamo questo metodo. Questa sezione funge da riferimento per le implementazioni che utilizzano versioni precedenti.
- Questo metodo non è compatibile con [Allocazione di più tipi di istanza con Slurm](#).

1. Crea un gruppo di risorse, per raggruppare la capacità.

```

$ aws resource-groups create-group --name EC2CRGroup \
 --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}

```

**Note**

Un gruppo di risorse non supporta risorse condivise da altri account.

Se l'ODCR di destinazione è condiviso da un altro account, non è necessario creare un gruppo di risorse. Utilizza `CapacityReservationId` al posto di un gruppo di risorse nel passaggio 3.

```
#!/bin/bash
set -e

Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationId": "cr-abcdef01234567890"
 }
 }
 }
 }
}
EOF
```

Aggiungi prenotazioni di capacità al gruppo di risorse. Ogni volta che crei un nuovo ODCR, aggiungilo alla prenotazione di gruppo. Sostituiscilo `ACCOUNT_ID` con l'ID del tuo account, `PLACEHOLDER_CAPACITY_RESERVATION` con l'ID di prenotazione della capacità e `REGION_ID` con il tuo Regione AWS ID (ad esempio, us-east-1).

```
$ aws resource-groups group-resources --region REGION_ID --group EC2CRGroup \
 --resource-arns arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-
 reservation/PLACEHOLDER_CAPACITY_RESERVATION
```

Crea un documento di policy sul tuo computer locale. Sostituiscilo `ACCOUNT_ID` con l'ID del tuo account e `REGION_ID` con il tuo Regione AWS ID (ad esempio, us-east-1).

```
cat > policy.json << EOF
```

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "RunInstancesInCapacityReservation",
 "Effect": "Allow",
 "Action": "ec2:RunInstances",
 "Resource": [
 "arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-reservation/*",
 "arn:aws:resource-groups:REGION_ID:ACCOUNT_ID:group/*"
]
 }
]
}
EOF
```

2. Crea la policy IAM Account AWS utilizzando il file json che hai creato.

```
$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-document file://policy.json
```

3. Crea il seguente script post-installazione localmente sull'istanza e assegnagli un **postinstall.sh** nome.

Sostituisci **ACCOUNT\_ID** con il tuo Account AWS ID e **REGION\_ID** con il tuo Regione AWS ID (ad esempio, us-east-1).

```
#!/bin/bash
set -e

Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:REGION_ID:ACCOUNT_ID:group/EC2CRGroup"
 }
 }
 }
 }
}
```

```
}
EOF
```

Caricare il file immagine in un bucket Amazon S3. *amzn-s3-demo-bucket* Sostituiscilo con il nome specifico del bucket S3.

```
$ aws s3 mb s3://amzn-s3-demo-bucket
aws s3 cp postinstall.sh s3://amzn-s3-demo-bucket/postinstall.sh
```

4. Crea la configurazione del cluster locale, sostituendo i segnaposto con i tuoi valori.

```
Region: REGION_ID
Image:
 Os: alinux2
HeadNode:
 InstanceType: c5.2xlarge
 Ssh:
 KeyName: YOUR_SSH_KEY
 Iam:
 S3Access:
 - BucketName: amzn-s3-demo-bucket
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::ACCOUNT_ID:policy/RunInstancesCapacityReservation
This post-install script is executed after the node is configured.
It is used to install scripts at boot time and specific configurations
In the script below we are overriding the calls to RunInstance to force
the provisioning of our my-queue partition to go through
the On-Demand Capacity Reservation
CustomActions:
 OnNodeConfigured:
 Script: s3://amzn-s3-demo-bucket/postinstall.sh
Networking:
 SubnetId: YOUR_PUBLIC_SUBNET_IN_TARGET_AZ

Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: my-queue
 ComputeResources:
 - MinCount: 0
 MaxCount: 100
 InstanceType: p4d.24xlarge
 Name: my-compute-resource
```

```

Efa:
 Enabled: true
Networking:
 ## PlacementGroup:
 ## Enabled: true ## Keep PG disabled if using targeted ODCR
 SubnetIds:
 - YOUR_PRIVATE_SUBNET_IN_TARGET_AZ

```

## 5. Crea il cluster.

Utilizzate il seguente comando per creare il cluster. Sostituisci *cluster-config.yaml* con il nome del file di configurazione, *cluster-dl* con il nome del cluster e *REGION\_ID* con il tuo ID regionale (ad esempio, us-east-1).

```

$ pcluster create-cluster --cluster-configuration cluster-config.yaml --cluster-
name cluster-dl --region REGION_ID

```

Dopo la creazione del cluster, lo script di post-installazione viene eseguito nel nodo principale. Lo script crea il `run_instances_overrides.json` file e sovrascrive le chiamate per `RunInstances` forzare il provisioning della partizione attraverso la On-Demand Capacity Reservation.

I AWS ParallelCluster daemon responsabili della scalabilità del cluster utilizzano automaticamente questa configurazione per le nuove istanze che vengono lanciate. Per confermare che i parametri specificati vengono utilizzati per il provisioning delle istanze, puoi consultare i seguenti file di registro:

- `/var/log/parallelcluster/clustermgtd(per capacità statica -) MinCount > 0`
- `/var/log/parallelcluster/slurm_resume.log(per capacità dinamica)`

Se i parametri sono corretti, troverai una voce di registro contenente quanto segue.

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

## Aggiornare le **RunInstances** sostituzioni

Puoi aggiornare la configurazione JSON generata in qualsiasi momento senza interrompere la flotta di elaborazione. Dopo l'applicazione delle modifiche, tutte le nuove istanze vengono avviate con la



configurazione aggiornata. Se devi applicare la configurazione aggiornata ai nodi in esecuzione, ricicla i nodi forzando la chiusura dell'istanza e attendi la sostituzione AWS ParallelCluster di tali nodi. Puoi farlo terminando l'istanza dalla console Amazon EC2 AWS CLI oppure impostando Slurm i nodi in DOWN uno DRAIN stato or.

Usa il comando seguente per impostare il Slurm nodo su oDOWN. DRAIN

```
$ scontrol update nodename=my-queue-dy-my-compute-resource-1 state=down
reason=your_reason
scontrol update nodename=my-queue-dy-my-compute-resource-1 state=drain
reason=your_reason
```

## Avvia istanze con Capacity Blocks (CB)

AWS ParallelCluster supporta [On-Demand Capacity Reservations \(ODCR\)](#) e [Capacity Blocks \(CB\) per Machine Learning](#). A differenza dell'ODCR, CB può avere orari di inizio futuri ed è limitato nel tempo. Per ulteriori informazioni sull'avvio con ODCR, consulta [Launch instances with On-Demand Capacity Reservations \(ODCR\)](#).

## Usare CB con AWS ParallelCluster

Per configurare i tuoi cluster nuovi o esistenti per utilizzare un CB, devi prima avere un CB valido nel tuo account. AWS Puoi usare Console di gestione AWS AWS Command Line Interface, o SDK per trovare e acquistare un CB disponibile seguendo la documentazione ufficiale. Una volta che hai un CB valido, puoi impostare il CB Amazon Resource Name (ARN) e i relativi parametri nel tuo file di configurazione. AWS ParallelCluster Per ulteriori informazioni, consulta [Find and purchase Capacity Blocks \(CB\)](#)

### CB nella configurazione del cluster

Per utilizzare un CB per una coda specifica è necessario utilizzare il parametro.

CapacityReservationId Configuralo su un ID CB esistente. È possibile ottenere l'ARN CB dal AWS CLI, o Console di gestione AWS dall'SDK utilizzato per creare il CB.

È necessario impostare CapacityType = CAPACITY\_BLOCK la coda in cui si desidera utilizzare il CB. Impostalo sulla risorsa InstanceType di calcolo (la stessa del tipo di istanza Amazon Elastic Compute Cloud del CB).

Quando lo specifichi CapacityReservationId a livello di risorsa di calcolo, InstanceType è facoltativo perché verrà recuperato automaticamente dalla prenotazione.

Quando si utilizza `CapacityType = CAPACITY_BLOCK`, `MaxCount` deve essere uguale `MinCount` e maggiore di 0, poiché tutte le istanze che fanno parte della prenotazione CB sono gestite come nodi statici.

Al momento della creazione del cluster, il nodo principale attende che tutti i nodi statici siano pronti prima di segnalare il successo della creazione del cluster. Tuttavia, quando si utilizza `CapacityType = CAPACITY_BLOCK`, i nodi che fanno parte delle risorse di elaborazione associate non verranno presi in considerazione per questo controllo. Il cluster verrà creato anche se tutti i dati configurati non sono attivi.

Il seguente frammento di file di configurazione mostra i parametri necessari per l'abilitazione nel file di AWS ParallelCluster configurazione.

```
SlurmQueues:
- Name: string
 CapacityType: CAPACITY_BLOCK
 ComputeResources:
 - Name: string
 InstanceType: String (EC2 Instance type of the CB)
 MinCount: integer (<= total capacity of the CB)
 MaxCount: integer (equal to MinCount)
 CapacityReservationTarget:
 CapacityReservationId: String (CB id)
```

## Come AWS ParallelCluster utilizza Capacity Blocks (CB)

AWS ParallelCluster gestisce i nodi statici associati in un modo particolare. AWS ParallelCluster crea un cluster anche se il CB non è ancora attivo e le istanze vengono avviate automaticamente una volta che il CB è attivo.

I Slurm nodi che corrispondono alle risorse di calcolo, sono associati e non sono ancora attivi, vengono mantenuti in manutenzione fino al raggiungimento dell'ora di avvio del CB. Slurmi nodi rimangono in uno reservation/maintenance stato e sono associati all'utente amministratore di slurm. Ciò significa che possono accettare lavori, ma i lavori rimangono pending fino a quando la prenotazione non viene rimossa.

AWS ParallelCluster aggiorna automaticamente Slurm le prenotazioni e mette i relativi nodi CB in stato di manutenzione (corrispondente allo stato CB). Quando il CB è attivo, la Slurm prenotazione viene rimossa e i nodi si avviano e diventano disponibili per i lavori in sospeso o per l'invio di nuovi lavori.

Quando viene raggiunta l'ora di fine del CB, i nodi verranno riportati a uno stato. `reservation/maintenance` Spetta agli utenti eseguire `resubmit/requeue` i lavori su una nuova coda/risorsa di calcolo quando il CB non è più attivo e le istanze vengono terminate.

## Patch AMI e sostituzione delle EC2 istanze Amazon

Per garantire che tutti i nodi di calcolo del cluster avviati dinamicamente si comportino in modo coerente, AWS ParallelCluster disabilita gli aggiornamenti automatici del sistema operativo delle istanze di cluster. Inoltre, viene creato un set specifico per ogni versione AWS ParallelCluster e la CLI associata. AWS ParallelCluster AMIs Questo set specifico AMIs rimane invariato e viene supportato solo dalla AWS ParallelCluster versione per cui è stato creato. AWS ParallelCluster AMIs perché le versioni rilasciate non sono aggiornate.

Tuttavia, a causa di problemi di sicurezza emergenti, i clienti potrebbero voler aggiungere patch a queste AMIs e quindi aggiornare i propri cluster con l'AMI con patch. [Ciò è in linea con il modello di responsabilità condivisa.](#) [AWS ParallelCluster](#)

Per visualizzare il set specifico AWS ParallelCluster AMIs supportato dalla versione AWS ParallelCluster CLI che stai attualmente utilizzando, esegui:

```
$ pcluster version
$ pcluster list-official-images
```

Il nodo AWS ParallelCluster principale è un'istanza statica e puoi aggiornarlo manualmente. Il riavvio e il riavvio del nodo principale sono completamente supportati a partire dalla AWS ParallelCluster versione 3.0.0.

Se le istanze dispongono di istanze store temporanei, è necessario ricordarsi di salvare i dati dell'instance store prima degli aggiornamenti manuali. Per ulteriori informazioni, consulta la configurazione [HeadNodeLocalStorage//EphemeralVolume](#) cluster e i [tipi di istanza con volumi di archivio delle istanze](#) nella Amazon EC2 User Guide for Linux Instances.

I nodi di calcolo sono istanze effimere. Per impostazione predefinita, puoi accedervi solo dal nodo principale. A partire dalla AWS ParallelCluster versione 3.0.0, puoi aggiornare l'AMI associata alle istanze di calcolo modificando il [CustomAmi](#) parametro [Scheduling//SlurmQueuesImage](#)/ed eseguendo il [pcluster update-cluster](#) comando, dopo aver interrotto il parco di calcolo con: [pcluster update-compute-fleet](#)

```
$ pcluster update-compute-fleet-status --status STOP_REQUESTED
```

È possibile automatizzare la creazione di un'AMI personalizzata aggiornata per i nodi di calcolo utilizzando uno dei seguenti metodi:

- Usa il [pcluster build-image](#) comando con una [Build](#)/aggiornata. [ParentImage](#)
- Esegui la build con [BuildUpdateOsPackages/Enabled](#):true.

## Aggiornamento o sostituzione dell'istanza del nodo principale

In alcune circostanze, potrebbe essere necessario riavviare o riavviare il nodo principale. Ad esempio, è necessario quando si aggiorna manualmente il sistema operativo o quando è [previsto il ritiro pianificato di un'AWS istanza](#) che impone il riavvio dell'istanza del nodo principale.

Se la tua istanza non dispone di unità temporanee, puoi interromperla e riavviarla in qualsiasi momento. Nel caso di un ritiro programmato, l'avvio dell'istanza interrotta la migra per utilizzare il nuovo hardware.

Allo stesso modo, puoi interrompere e avviare manualmente un'istanza che non dispone di archivi di istanze. In questo caso e in altri casi di istanze senza volumi effimeri, continua con. [Arresta e avvia il nodo principale di un cluster](#)

Se l'istanza ha unità temporanee ed è stata interrotta, i dati nell'instance store andranno persi. È possibile determinare se il tipo di istanza utilizzato per il nodo principale dispone di archivi di istanze dalla tabella disponibile nei volumi di [Instance](#) store.

## Salva i dati da unità temporanee

A partire dalla AWS ParallelCluster versione 3.0.0, il riavvio e il riavvio del nodo principale sono completamente supportati per ogni tipo di istanza. Tuttavia, se le istanze dispongono di un'unità temporanea, i relativi dati vengono persi. Segui i passaggi successivi per conservare i dati prima del riavvio o del riavvio del nodo principale.

Per verificare se hai dati che devono essere conservati, visualizza il contenuto nella [MountDir](#)cartella [EphemeralVolume](#)/(/scratchper impostazione predefinita).

Puoi trasferire i dati sul volume root o sui sistemi di storage condivisi collegati al cluster, come Amazon FSx, Amazon EFS o Amazon EBS. Tieni presente che il trasferimento dei dati verso lo storage remoto può comportare costi aggiuntivi.

Dopo aver salvato i dati, continua con. [Arresta e avvia il nodo principale di un cluster](#)

## Arresta e avvia il nodo principale di un cluster

1. Verifica che non ci siano processi in esecuzione nel cluster.

Quando si utilizza un Slurm programmatore:

- Se l'`sbatch--no-requeue` opzione non è specificata, vengono richiesti i lavori in esecuzione.
- Se l'`--no-requeue` opzione è specificata, i processi in esecuzione hanno esito negativo.

2. Richiedi un'interruzione della flotta di elaborazione del cluster:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
 "status": "STOP_REQUESTED",
 ...
}
```

3. Attendi che lo stato della flotta di elaborazione sia: STOPPED

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
 "status": "STOPPED",
 ...
}
```

4. Per gli aggiornamenti manuali con il riavvio del sistema operativo o il riavvio dell'istanza, puoi utilizzare o. Console di gestione AWS AWS CLI Di seguito è riportato un esempio di utilizzo di AWS CLI.

```
Retrieve head node instance id
$ pcluster describe-cluster --cluster-name cluster-name --status STOP_REQUESTED
{
 "headNode": {
 "instanceId": "i-1234567890abcdef0",
 ...
 },
 ...
}
stop and start the instance
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
 "StoppingInstances": [
```

```

 {
 "CurrentState": {
 "Name": "stopping"
 ...
 },
 "InstanceId": "i-1234567890abcdef0",
 "PreviousState": {
 "Name": "running"
 ...
 }
 }
]
}
$ aws ec2 start-instances --instance-ids 1234567890abcdef0
{
 "StartingInstances": [
 {
 "CurrentState": {
 "Name": "pending"
 ...
 },
 "InstanceId": "i-1234567890abcdef0",
 "PreviousState": {
 "Name": "stopped"
 ...
 }
 }
]
}

```

##### 5. Avvia la flotta di elaborazione del cluster:

```

$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
 "status": "START_REQUESTED",
 ...
}

```

# Sistemi operativi

AWS ParallelCluster supporta Amazon Linux 2, Amazon Linux 2023, Ubuntu24.04, Ubuntu 22.04, Ubuntu 20.04, Red Hat Enterprise Linux 8 (RHEL8), Rocky 8, Red Hat Enterprise Linux 9 () e Rocky 9. RHEL9 AWS ParallelCluster offre funzionalità predefinite AMIs per determinati sistemi operativi, per maggiori dettagli su quelle fornite da. AMIs AWS ParallelCluster [Sezione Image](#)

## Considerazioni sul sistema operativo

### Ubuntu 22.04 e Ubuntu 24.04

Ubuntu 22.04 e Ubuntu 24.04 richiedono chiavi più sicure per ssh e non supportano le chiavi RSA per impostazione predefinita. Genera una chiave ed25519 e usala per la creazione del cluster.

Ubuntu 22.04 non può essere aggiornato all'ultimo kernel perché non esiste un FSx client per quel kernel.

### RHEL 8

RedHat Enterprise Linux 8.7 (rhel8) viene aggiunto a partire dalla versione 3.6.0. AWS ParallelCluster Se si configura il cluster per utilizzare rhel8, il costo on-demand per qualsiasi tipo di istanza è superiore rispetto a quando si configura il cluster per utilizzare altri sistemi operativi supportati.

Per ulteriori informazioni sui prezzi, consulta la sezione Prezzi [On-Demand e In che modo viene offerto e prezzato Red Hat Enterprise Linux su Amazon Elastic Compute Cloud?](#) .

### Rocky 8

AWS ParallelCluster 3.8.0 supporta Rocky Linux 8, ma Rocky Linux 8 preconfigurato AMIs (per architetture x86 e ARM) non è disponibile. AWS ParallelCluster 3.8.0 supporta la creazione di cluster con Rocky Linux 8 utilizzando la proprietà personalizzata. AMIs [CustomAmi](#) Per ulteriori informazioni sulla creazione di soluzioni personalizzate AMIs, fare riferimento a. [AWS ParallelCluster Personalizzazione AMI](#)

[Per creare la tua AMI personalizzata da un'AMI Rocky Linux 8 di base, puoi prendere in considerazione l'abbonamento a Rocky Linux 8 AMIs disponibile sul Marketplace. AWS](#) Assicurati di controllare i prezzi e i costi di abbonamento per Rocky Linux 8 AMIs sul AWS Marketplace. In alternativa puoi anche usare [Rocky Linux 8 ufficiale AMIs](#) come AMI di base.

### Rocky 9

AWS ParallelCluster 3.9.0 supporta Rocky Linux 9, ma Rocky Linux 9 preconfigurato AMIs (per architetture x86 e ARM) non è disponibile. AWS ParallelCluster 3.9.0 supporta la creazione di cluster con Rocky Linux 9 utilizzando la proprietà personalizzata. AMIs [CustomAmi](#) Per ulteriori informazioni sulla creazione personalizzata AMIs, consulta la sezione [Personalizzazione AWS ParallelCluster AMI](#). Per creare la tua AMI personalizzata da una AMI Rocky Linux 9 di base, puoi anche usare [Rocky Linux 9 ufficiale AMIs](#) come AMI di base. La compilazione personalizzata dell'AMI Rocky Linux 9 potrebbe fallire se l'AMI di base non ha il kernel più recente. Per aggiornare il kernel prima di creare l'AMI:

- [Avvia un'istanza utilizzando un ID AMI rocky9 da qui: https://rockylinux.org/cloud-images/](https://rockylinux.org/cloud-images/)
- ssh nell'istanza ed esegui il seguente comando: `sudo yum -y update`
- Crea un'immagine dall'istanza da usare come ParentImage



# Riferimento per AWS ParallelCluster

## Argomenti

- [AWS ParallelCluster Comandi CLI](#)
- [File di configurazione](#)
- [AWS ParallelCluster Riferimento API](#)
- [AWS ParallelCluster API della libreria Python](#)

## AWS ParallelCluster Comandi CLI

`pcluster` è il comando AWS ParallelCluster CLI principale. È possibile `pcluster` avviare e gestire cluster HPC e creare Cloud AWS e gestire immagini AMI personalizzate.

`pcluster3-config-converter` viene utilizzato per convertire le configurazioni dei cluster nel formato AWS ParallelCluster versione 2 nel formato AWS ParallelCluster versione 3.

```
pcluster [-h] (build-image | configure |
 create-cluster | dcv-connect |
 delete-cluster | delete-cluster-instances | delete-image |
 describe-cluster | describe-cluster-instances |
 describe-compute-fleet | describe-image |
 export-cluster-logs | export-image-logs |
 get-cluster-log-events | get-cluster-stack-events |
 get-image-log-events | get-image-stack-events |
 list-cluster-log-streams | list-clusters |
 list-images | list-image-log-streams | list-official-images |
 ssh | update-cluster |
 update-compute-fleet | version) ...
pcluster3-config-converter [-h] [-t CLUSTER_TEMPLATE]
 [-c CONFIG_FILE]
 [--force-convert]
 [-o OUTPUT_FILE]
```

## Argomenti

- [pcluster](#)
- [pcluster3-config-converter](#)

# pcluster

pcluster è il comando AWS ParallelCluster CLI principale. Viene utilizzato pcluster per avviare e gestire i cluster HPC in Cloud AWS.

pcluster scrive i registri dei comandi nei file in `pcluster.log`. # `/home/user/.parallelcluster/` Per ulteriori informazioni, consulta [pclusterRegistri CLI](#).

Per utilizzarlo pcluster, devi disporre di un ruolo IAM con le [autorizzazioni](#) necessarie per eseguirlo.

```
pcluster [-h]
```

## Arguments (Argomenti)

### pcluster *command*

Scelte possibili: [build-image](#)[configure](#)[create-cluster](#)| [dcv-connect](#)| [delete-cluster](#)| [delete-cluster-instances](#)| [delete-image](#)| [describe-cluster](#)| [describe-cluster-instances](#)| [describe-compute-fleet](#)| [describe-image](#)| [export-cluster-logs](#)| [export-image-logs](#)| [get-cluster-log-events](#)| [get-cluster-stack-events](#)| [get-image-log-events](#)| [get-image-stack-events](#)| [list-clusters](#)[list-cluster-log-streams](#)| [list-images](#)| [list-image-log-streams](#)[list-official-images](#)| [ssh](#)| [update-cluster](#)| [update-compute-fleet](#)| [version](#)

Sottocomandi:

Argomenti

- [pcluster build-image](#)
- [pcluster configure](#)
- [pcluster create-cluster](#)
- [pcluster dcv-connect](#)
- [pcluster delete-cluster](#)
- [pcluster delete-cluster-instances](#)
- [pcluster delete-image](#)
- [pcluster describe-cluster](#)

- [pcluster describe-cluster-instances](#)
- [pcluster describe-compute-fleet](#)
- [pcluster describe-image](#)
- [pcluster export-cluster-logs](#)
- [pcluster export-image-logs](#)
- [pcluster get-cluster-log-events](#)
- [pcluster get-cluster-stack-events](#)
- [pcluster get-image-log-events](#)
- [pcluster get-image-stack-events](#)
- [pcluster list-clusters](#)
- [pcluster list-cluster-log-streams](#)
- [pcluster list-images](#)
- [pcluster list-image-log-streams](#)
- [pcluster list-official-images](#)
- [pcluster ssh](#)
- [pcluster update-cluster](#)
- [pcluster update-compute-fleet](#)
- [pcluster version](#)

## pcluster build-image

Crea un' AWS ParallelCluster immagine personalizzata nella regione specificata.

```
pcluster build-image [-h]
 --image-configuration IMAGE_CONFIGURATION
 --image-id IMAGE_ID
 [--debug]
 [--dryrun DRYRUN]
 [--query QUERY]
 [--region REGION]
 [--rollback-on-failure ROLLBACK_ON_FAILURE]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```

## Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster build-image`.

### **--image-configuration, -c *IMAGE\_CONFIGURATION***

Specifica il file di configurazione dell'immagine come documento YAML.

### **--image-id, -i *IMAGE\_ID***

Specifica l'id dell'immagine che verrà creata.

### **--debug**

Attiva la registrazione di debug.

### **--dryrun *DRYRUN***

Quando `true`, il comando esegue la convalida senza creare risorse. È possibile utilizzarlo per convalidare la configurazione dell'immagine. (Il valore predefinito è.) `false`

### **--query *QUERY***

JMESPath interrogazione da eseguire in uscita.

### **--region, -r *REGION***

Specifica Regione AWS da usare. Regione AWS Deve essere specificato utilizzando l'impostazione [Region](#) nel file di configurazione dell'immagine, la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

Quando `true`, avvia automaticamente un rollback dello stack di immagini in caso di errore. (Il valore predefinito è.) `false`

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

Identifica uno o più validatori di configurazione da sopprimere.

Formato: `(|) ALL type:[A-Za-z0-9]+`

### **--validation-failure-level {INFO,WARNING,ERROR}**

Specifica il livello di convalida minimo che impedirà la creazione. (Il valore predefinito è.) `ERROR`

## Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster build-image --image-configuration image-config.yaml --image-id custom-alinux2-image
{
 "image": {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/custom-alinux2-image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.2"
 }
}
```

### Warning

`pcluster build-image` utilizza il VPC predefinito. Se il VPC predefinito è stato eliminato, magari utilizzando AWS Control Tower o AWS Landing Zone, l'ID della sottorete deve essere specificato nel file di configurazione dell'immagine. Per ulteriori informazioni, consulta [SubnetId](#).

## pcluster configure

Avvia una procedura guidata di configurazione interattiva per la AWS ParallelCluster versione 3. Per ulteriori informazioni, consulta [Configura e crea un cluster con AWS ParallelCluster interfaccia a riga di comando](#).

```
pcluster configure [-h]
 --config CONFIG
 [--debug]
 [--region REGION]
```

Argomenti denominati

**-h, --help**

Mostra il testo di aiuto per `pcluster configure`.

**--config *CONFIG***

Percorso per generare il file di configurazione generato.

**--debug**

Attiva la registrazione di debug.

**--region, -r *REGION***

Specifica Regione AWS da usare. La regione deve essere specificata utilizzando l'impostazione [Region](#) nel file di configurazione dell'immagine, la variabile di AWS\_DEFAULT\_REGION ambiente, l'regionimpostazione nella [default] sezione del ~/.aws/config file o il --region parametro.

**pcluster create-cluster**

Crea un AWS ParallelCluster cluster.

```
pcluster create-cluster [-h]
 --cluster-configuration CLUSTER_CONFIGURATION
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--query QUERY]
 [--region REGION]
 [--rollback-on-failure ROLLBACK_ON_FAILURE]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```

Argomenti denominati

**-h, --help**

Mostra il testo di aiuto perpcluster create-cluster.

**--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

Specifica il file di configurazione del cluster YAML.

**--cluster-name, -n *CLUSTER\_NAME***

Speciifica il nome del cluster da creare.

Il nome deve iniziare con un carattere alfabetico. Il nome può contenere fino a 60 caratteri. Se la Slurm contabilità è abilitata, il nome può contenere fino a 40 caratteri.

Caratteri validi: a-z, A-Z, 0-9 e - (trattino).

### **--debug**

Abilita la registrazione del debug.

### **--dryrun *DRYRUN***

Quando `true`, il comando esegue la convalida senza creare risorse. È possibile utilizzarlo per convalidare la configurazione del cluster. (Il valore predefinito è.) `false`

### **--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

### **--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando l'[Region](#) impostazione nel file di configurazione del cluster, la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

Quando `true`, avvia automaticamente un rollback dello stack del cluster in caso di errori. (Il valore predefinito è.) `true`

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS* ...]**

Identifica uno o più validatori di configurazione da sopprimere.

Formato: (`[type:] ALL [A-Za-z0-9]+`)

### **--validation-failure-level {INFO,WARNING,ERROR}**

Specifica il livello di convalida minimo che impedirà la creazione. (Il valore predefinito è.) `ERROR`

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster create-cluster -c cluster-config.yaml -n cluster-v3
{
 "cluster": {
```

```
{
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_IN_PROGRESS"
}
```

## pcluster dcv-connect

Consente di connettersi al nodo principale tramite una sessione interattiva utilizzando Amazon DCV.

```
pcluster dcv-connect [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--key-path KEY_PATH]
 [--login-node-ip LOGIN_NODE_IP]
 [--region REGION]
 [--show-url]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster dcv-connect`.

### --cluster-name, -n *CLUSTER\_NAME*

Specificate il nome del cluster.

### --debug

Abilita la registrazione del debug.

### --key-path *KEY\_PATH*

Specifica il percorso della chiave SSH da utilizzare per la connessione.

### --login-node-ip

Specifica l'indirizzo IP pubblico o privato di un nodo di accesso nel cluster. L'utilizzo di questo argomento consente la connessione a un nodo di accesso nel cluster con DCV abilitato.



**Note**

Questo argomento è stato aggiunto nella AWS ParallelCluster versione 3.11.0.

**--region, -r *REGION***

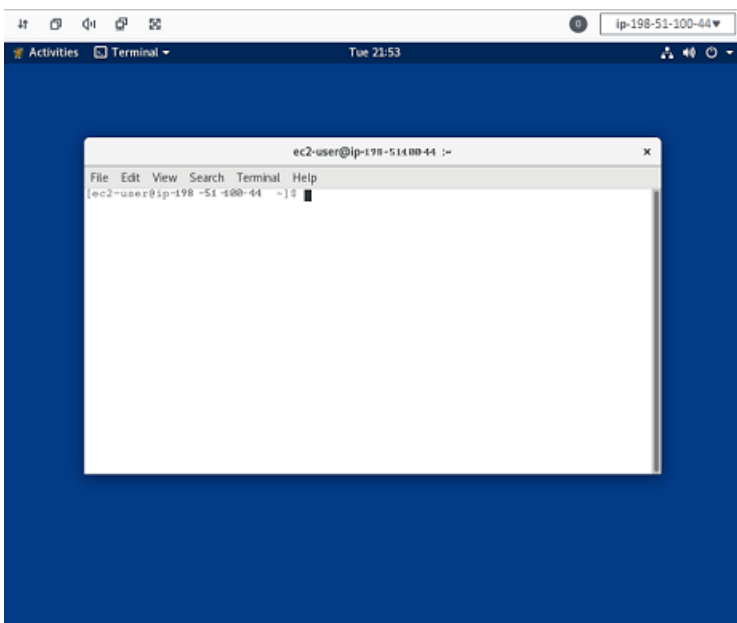
Specifica l'uso. Regione AWS Regione AWS Deve essere specificato utilizzando la variabile di AWS\_DEFAULT\_REGION ambiente, l'regionimpostazione nella [default] sezione del ~/.aws/config file o il --region parametro.

**--show-url**

Stampa l'URL che verrebbe utilizzato per la connessione e le uscite DCV.

Esempio di utilizzo AWS ParallelCluster della versione 3.11

```
$ pcluster dcv-connect -n cluster-3Dcv --login-node-ip 198.51.100.44 -r us-east-1 --
key-path /home/user/.ssh/key.pem
```

**pcluster delete-cluster**

Avvia l'eliminazione di un cluster.

```
pcluster delete-cluster [-h]
```

```
--cluster-name CLUSTER_NAME
[--debug]
[--query QUERY]
[--region REGION]
```

Argomenti denominati

## **-h, --help**

Mostra il testo di aiuto per `pcluster delete-cluster`.

## **--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

## **--debug**

Abilita la registrazione del debug.

## **--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

## **--region, -r *REGION***

Specifica l'uso Regione AWS . La regione deve essere specificata utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster delete-cluster -n cluster-v3
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

## pcluster delete-cluster-instances

Avvia la chiusura forzata di tutti i nodi di calcolo del cluster. Questa operazione non funziona con i cluster AWS Batch

```
pcluster delete-cluster-instances [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--force FORCE]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster delete-cluster-instances`.

### --cluster-name, -n *CLUSTER\_NAME*

Specificate il nome del cluster.

### --debug

Abilita la registrazione del debug.

### --force *FORCE*

When `true`, forza l'eliminazione ignorando gli errori di convalida. (Il valore predefinito è.) `false`

### --query *QUERY*

Specifica l' JMESPath interrogazione da eseguire sull'output.

### --region, -r *REGION*

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

```
$ pcluster delete-cluster-instances -n cluster-v3
```

## pcluster delete-image

Avvia l'eliminazione dell' AWS ParallelCluster immagine personalizzata.

```
pcluster delete-image [-h]
 --image-id IMAGE_ID
 [--debug]
 [--force FORCE]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster delete-image`.

### **--image-id, -i *IMAGE\_ID***

Specifica l'id dell'immagine che verrà eliminata.

### **--debug**

Abilita la registrazione del debug.

### **--force *FORCE***

Quando `true`, forza l'eliminazione nel caso in cui vi siano istanze che utilizzano l'AMI o se l'AMI è condiviso. (Il valore predefinito è.) `false`

### **--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

### **--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'regionimpostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster delete-image --image-id custom-alinux2-image
{
 "image": {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "DELETE_IN_PROGRESS",
 "region": "us-east-1",
 "version": "3.1.4"
```

```
}
}
```

## pcluster describe-cluster

Ottieni informazioni dettagliate su un cluster.

```
pcluster describe-cluster [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster describe-cluster`.

### --cluster-name, -n *CLUSTER\_NAME*

Specificate il nome del cluster.

### --debug

Abilita la registrazione del debug.

### --query *QUERY*

Specifica l' JMESPath interrogazione da eseguire sull'output.

### --region, -r *REGION*

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempi che utilizzano AWS ParallelCluster la versione 3.1.4:

Descrivi i dettagli del cluster:

```
$ pcluster describe-cluster -n cluster-v3
{
 "creationTime": "2022-07-12T17:19:16.101Z",
 "headNode": {
```

```

 "launchTime": "2022-07-12T17:22:21.000Z",
 "instanceId": "i-1234567890abcdef0",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "running",
 "privateIpAddress": "192.0.2.0.196"
 },
 "loginNodes": [
 {
 "status": "active",
 "poolName": "pool1",
 "address": "cluster-v3-eMr9BYRKZVda-e5bb34f40b24f51d.elb.us-
east-1.amazonaws.com",
 "scheme": "internet-facing",
 "healthyNodes": 1,
 "unhealthyNodes": 0
 },
 {
 "status": "active",
 "poolName": "pool2",
 "address": "cluster-v3-PaQ7GgC27sic-aba10c890247b36b.elb.us-
east-1.amazonaws.com",
 "scheme": "internet-facing",
 "healthyNodes": 1,
 "unhealthyNodes": 0
 }
],
 "version": "3.1.4",
 "clusterConfiguration": {
 "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
 },
 "tags": [
 {
 "value": "3.11",
 "key": "parallelcluster:version"
 }
],
 "cloudFormationStackStatus": "CREATE_COMPLETE",
 "clusterName": "cluster-v3",
 "computeFleetStatus": "RUNNING",
 "cloudFormationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "lastUpdatedTime": "2022-07-12T17:19:16.101Z",
 "region": "us-east-1",

```

```
"clusterStatus": "CREATE_COMPLETE"
}
```

Utilizzare `describe-cluster` per recuperare la configurazione del cluster:

```
$ curl -o - - $(pcluster describe-cluster -n cluster-v3 --query clusterConfiguration.url
| xargs echo)
Region: us-east-1
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: adpc
 Iam:
 S3Access:
 - BucketName: cluster-v3-bucket
 KeyName: logs
 EnableWriteAccess: true
 Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue1
 ComputeResources:
 - Name: t2micro
 InstanceType: t2.micro
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-021345abcdef6789
```

## **pcluster describe-cluster-instances**

Descrivi le istanze in un cluster.

```
pcluster describe-cluster-instances [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--next-token NEXT_TOKEN]
 [--node-type {HeadNode,ComputeNode,LoginNode}]
```

```
[--query QUERY]
[--queue-name QUEUE_NAME]
[--region REGION]
```

Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster describe-cluster-instances`.

### **--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

### **--debug**

Abilita la registrazione del debug.

### **--next-token *NEXT\_TOKEN***

Token per il set di risultati successivo.

### **--node-type {HeadNode,ComputeNode,LoginNode}**

Specifica i tipi di nodo da elencare. I valori supportati sono `HeadNode`, `ComputeNode` e `LoginNode`. Se questo parametro non è specificato, vengono descritte le istanze `HeadNode`, `ComputeNode` e `LoginNode`.

### **--query *QUERY***

Specifica l' `JMESPath` interrogazione da eseguire sull'output.

### **--queue-name *QUEUE\_NAME***

Specifica il nome della coda da elencare. Se questo parametro non è specificato, vengono descritte le istanze presenti in tutte le code.

### **--region, -r *REGION***

Specifica l'uso. Regione AWS Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:



```
$ pcluster describe-cluster-instances -n cluster-v3
{
 "instances": [
 {
 "launchTime": "2022-07-12T17:22:21.000Z",
 "instanceId": "i-1234567890abcdef0",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "running",
 "nodeType": "HeadNode",
 "privateIpAddress": "192.0.2.0.196"
 },
 {
 "launchTime": "2022-07-12T17:37:42.000Z",
 "instanceId": "i-021345abcdef6789",
 "queueName": "queue1",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "pending",
 "nodeType": "ComputeNode",
 "privateIpAddress": "192.0.2.0.196"
 },
 {
 "launchTime": "2022-07-12T17:37:42.000Z",
 "instanceId": "i-021345abcdef6789",
 "poolName": "pool1",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "pending",
 "nodeType": "loginNode",
 "privateIpAddress": "192.0.2.0.196"
 }
]
}
```

## pcluster describe-compute-fleet

Descrivi lo stato della flotta di elaborazione.

```
pcluster describe-compute-fleet [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
```

```
[--region REGION]
```

Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster describe-compute-fleet`.

### **--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

### **--debug**

Abilita la registrazione del debug.

### **--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

### **--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'regionimpostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
 "status": "RUNNING",
 "lastStatusUpdatedTime": "2022-07-12T17:24:26.000Z"
}
```

## **pcluster describe-image**

Ottieni informazioni dettagliate su un'immagine.

```
pcluster describe-image [-h]
 --image-id IMAGE_ID
 [--debug]
 [--query QUERY]
 [--region REGION]
```

## Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster describe-image`.

### **--image-id, -i *IMAGE\_ID***

Specifica l'ID dell'immagine.

### **--debug**

Abilita la registrazione del debug.

### **--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

### **--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster describe-image --image-id custom-alinux2-image
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678-v1-do-not-delete.../configs/image-
config.yaml"
 },
 "imageId": "custom-alinux2-image",
 "creationTime": "2022-04-05T20:23:07.000Z"
 "imageBuildStatus": "BUILD_COMPLETE",
 "region": "us-east-1",
 "ec2AmiInfo": {
 "amiName": "custom-alinux2-image 2022-04-05T19-55-22.518Z",
 "amiId": "ami-1234abcd5678efgh",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.268-205.500.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.14.2-1.amzn2.x86_64, dcv-2021.3.11591-1.el7.x86_64, slurm-21-08-6-1",
 "state": "AVAILABLE",
 "tags": [
 {
```

```

 "value": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
 "key": "Ec2ImageBuilderArn"
 },
 {
 "value": "parallelcluster-1234abcd5678efgh-v1-do-not-delete",
 "key": "parallelcluster:amzn-s3-demo-bucket"
 },
 {
 "value": "custom-alinux2-image",
 "key": "parallelcluster:image_name"
 },
 {
 "value": "available",
 "key": "parallelcluster:build_status"
 },
 {
 "value": "s3://amzn-s3-demo-bucket/parallelcluster/3.1.2/images/custom-alinux2-
image-1234abcd5678efgh/configs/image-config.yaml",
 "key": "parallelcluster:build_config"
 },
 {
 "value": "Amazon EC2 Image Builder",
 "key": "CreatedBy"
 },
 {
 "value": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-alinux2-image",
 "key": "parallelcluster:build_log"
 },
 {
 "value": "4.14.268-205.500.amzn2.x86_64",
 "key": "parallelcluster:kernel_version"
 },
 {
 "value": "arn:aws:imagebuilder:us-east-1:444455556666:image/amazon-linux-2-
x86/2022.3.16/1",
 "key": "parallelcluster:parent_image"
 },
 {
 "value": "3.1.2",
 "key": "parallelcluster:version"
 },
 {

```

```
 "value": "0.5.14",
 "key": "parallelcluster:munge_version"
 },
 {
 "value": "21-08-6-1",
 "key": "parallelcluster:slurm_version"
 },
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 {
 "value": "alinux2-image",
 "key": "parallelcluster:image_id"
 },
 {
 "value": "3.2.3",
 "key": "parallelcluster:pmix_version"
 },
 {
 "value": "parallelcluster/3.15.0/images/alinux2-image-abcd1234efgh56781234",
 "key": "parallelcluster:s3_image_dir"
 },
 {
 "value": "1.14.2-1.amzn2.x86_64",
 "key": "parallelcluster:efa_version"
 },
 {
 "value": "alinux2",
 "key": "parallelcluster:os"
 },
 {
 "value": "aws-parallelcluster-cookbook-3.1.2",
 "key": "parallelcluster:bootstrap_file"
 },
 {
 "value": "1.8.23-10.amzn2.1.x86_64",
 "key": "parallelcluster:sudo_version"
 },
 {
 "value": "2.10.8-5.amzn2.x86_64",
 "key": "parallelcluster:lustre_version"
 }
],
```

```
"architecture": "x86_64"
},
"version": "3.1.2"
}
```

## pcluster export-cluster-logs

Esporta i log del cluster in un tar.gz archivio locale passando attraverso un Amazon S3 Bucket.

```
pcluster export-cluster-logs [-h]
 --cluster-name CLUSTER_NAME
 [--bucket BUCKET_NAME]
 [--bucket-prefix BUCKET_PREFIX]
 [--debug]
 [--end-time END_TIME]
 [--filters FILTER [FILTER ...]]
 [--keep-s3-objects KEEP_S3_OBJECTS]
 [--output-file OUTPUT_FILE]
 [--region REGION]
 [--start-time START_TIME]
```

### Note

Il `export-cluster-logs` comando attende che CloudWatch Logs completi l'esportazione dei log, quindi si prevede che trascorra un periodo di tempo senza alcun output.

Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster export-cluster-logs`.

### **--bucket *BUCKET\_NAME***

Specifica il nome del bucket Amazon S3 in cui esportare i dati dei log del cluster. Deve trovarsi nella stessa regione del cluster.

 Note

- È necessario aggiungere le autorizzazioni alla policy del bucket di Amazon S3 per concedere l'accesso. CloudWatch Per ulteriori informazioni, consulta [Impostare le autorizzazioni su un bucket Amazon S3](#) nella CloudWatch Logs User Guide.
- A partire dalla AWS ParallelCluster versione 3.12.0, l'opzione è facoltativa. **--bucket** Se l'opzione non è specificata, verrà utilizzato il bucket predefinito AWS ParallelCluster regionale (parallelcluster-hash-v1-D0-NOT-DELETE) oppure se il bucket Amazon S3 a CustomS3Bucket cui si fa riferimento è specificato nella configurazione del cluster, verrà utilizzato quello. Se non si specifica l'--bucketopzione e si utilizza il AWS ParallelCluster bucket predefinito, non è possibile esportare i log parallelcluster/ nella cartella, poiché si tratta di una cartella protetta riservata all'uso interno.

 Important

Se viene utilizzato il bucket AWS ParallelCluster predefinito, pcluster si occuperà di configurare la policy del bucket. Se hai personalizzato la policy del bucket e poi esegui l'aggiornamento alla AWS ParallelCluster versione 3.12.0, la policy del bucket verrà sovrascritta e dovrai riapplicare le modifiche.

**--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

**--bucket-prefix *BUCKET\_PREFIX***

Specifica il percorso nel bucket Amazon S3 in cui devono essere archiviati i dati dei log esportati.

Per impostazione predefinita, il prefisso bucket è:

```
cluster-name-logs-202209061743.tar.gz
```

*202209061743* è un esempio dell'ora in formato. %Y%m%d%H%M

 Note

A partire dalla AWS ParallelCluster versione 3.12.0, se non si specifica l'--bucketopzione e si utilizza il AWS ParallelCluster bucket predefinito, non è possibile esportare i registri `parallelcluster/` nella cartella, poiché si tratta di una cartella protetta riservata all'uso interno.

**--debug**

Abilita la registrazione di debug.

**--end-time *END\_TIME***

Specifica la fine dell'intervallo di tempo per la raccolta degli eventi di registro, espresso in formato ISO 8601 (YYYY-MM-DDThh:mm:ssZad esempio '). 2021-01-01T20:00:00Z Gli eventi con un timestamp uguale o successivo a questo orario non sono inclusi. Gli elementi temporali (ad esempio minuti e secondi) possono essere omessi. Il valore predefinito è l'ora corrente.

**--filters *FILTER* [*FILTER* ...]**

Specifica i filtri per il registro. Formato:Name=a,Values=1 Name=b,Values=2,3. I filtri supportati sono:

private-dns-name

Specifica la forma abbreviata del nome DNS privato dell'istanza (ad ip-10-0-0-101 esempio).

node-type

Specifica il tipo di nodo, l'unico valore accettato per questo filtro è. HeadNode

**--keep-s3-objects *KEEP\_S3\_OBJECTS***

Settrue, le esportazioni degli oggetti esportati in Amazon S3 vengono mantenute. (Il valore predefinito è.) false

**--output-file *OUTPUT\_FILE***

Specifica il percorso del file in cui salvare l'archivio di registro. Se viene fornito, i registri vengono salvati localmente. Altrimenti vengono caricati su Amazon S3 con l'URL restituito nell'output. L'impostazione predefinita è il caricamento su Amazon S3.



**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

**--start-time *START\_TIME***

Specifica l'inizio dell'intervallo di tempo, espresso nel formato ISO 8601 (ad `YYYY-MM-DDThh:mm:ssZ` esempio `2021-01-01T20:00:00Z`). Sono inclusi gli eventi di registro con un timestamp uguale a questo orario o successivo a tale ora. Se non specificato, l'impostazione predefinita è l'ora di creazione del cluster.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
{
 "url": "https://cluster-v3-bucket..."
}
```

Non riesci a recuperare i log?

Se non riesci a recuperare i log utilizzando il `export-cluster-logs` comando, esegui una delle seguenti operazioni:

- Recupera i log manualmente dal gruppo di CloudWatch log del cluster.
- Se il gruppo di log è vuoto, accedi tramite SSH ai nodi del cluster e recupera i log elencati in [Risoluzione dei problemi di inizializzazione dei nodi](#)
- Se i nodi del cluster non sono accessibili perché il cluster non è stato creato, ricrea il cluster con l'opzione `--rollback-on-failure false` e recupera i log dai nodi.

**pcluster export-image-logs**

Esporta i log dello stack di image builder in un `tar.gz` archivio locale passando attraverso un Amazon S3 Bucket.

```
pcluster export-image-logs [-h]
 --image-id IMAGE_ID
 [--bucket BUCKET]
```

```
[--bucket-prefix BUCKET_PREFIX]
[--debug]
[--end-time END_TIME]
[--keep-s3-objects KEEP_S3_OBJECTS]
[--output-file OUTPUT_FILE]
[--region REGION]
[--start-time START_TIME]
```

#### Note

Il `export-image-logs` comando attende che CloudWatch Logs completi l'esportazione dei log, quindi si prevede che trascorra un periodo di tempo senza alcun output.

## Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster export-image-logs`.

### **--bucket *BUCKET\_NAME***

Specifica il nome del bucket Amazon S3 in cui esportare i log di creazione dell'immagine. Deve trovarsi nella stessa regione dell'immagine.

#### Note

- È necessario aggiungere le autorizzazioni alla policy del bucket di Amazon S3 per concedere l'accesso. CloudWatch Per ulteriori informazioni, consulta [Impostare le autorizzazioni su un bucket Amazon S3](#) nella CloudWatch Logs User Guide.
- A partire dalla AWS ParallelCluster versione 3.12.0, l'opzione è facoltativa. **--bucket** Se l'opzione non è specificata, verrà utilizzato il bucket predefinito AWS ParallelCluster regionale (`parallelcluster-hash-v1-DO-NOT-DELETE`) oppure, se specificato nella configurazione dell'immagine, verrà utilizzato quello. `CustomS3Bucket`

#### Important

Se viene utilizzato il bucket AWS ParallelCluster predefinito, `pcluster` si occuperà di configurare la policy del bucket. Se personalizzi la policy del bucket prima di eseguire

l'aggiornamento alla AWS ParallelCluster versione 3.12.0, la policy del bucket verrà sovrascritta e dovrai riapplicare le modifiche.

**--image-id, -i *IMAGE\_ID***

L'ID dell'immagine i cui registri verranno esportati.

**--bucket-prefix *BUCKET\_PREFIX***

Specifica il percorso nel bucket Amazon S3 in cui devono essere archiviati i dati dei log esportati.

Per impostazione predefinita, il prefisso bucket è:

```
ami-id-logs-202209061743.tar.gz
```

*202209061743* è l'ora corrente nel formato. %Y%m%d%H%M

 **Note**

A partire dalla AWS ParallelCluster versione 3.12.0, se non si specifica l'--bucketopzione e si utilizza il AWS ParallelCluster bucket predefinito, non è possibile esportare i registri nella `parallelcluster/` cartella, poiché si tratta di una cartella protetta riservata all'uso interno.

**--debug**

Abilita la registrazione di debug.

**--end-time *END\_TIME***

Specifica la fine dell'intervallo di tempo per la raccolta degli eventi di registro, espresso in formato ISO 8601 (YYYY-MM-DDThh:mm:ssZad esempio '). 2021-01-01T20:00:00Z Gli eventi con un timestamp uguale o successivo a questo orario non sono inclusi. Gli elementi temporali (ad esempio minuti e secondi) possono essere omessi. Il valore predefinito è l'ora corrente.

**--keep-s3-objects *KEEP\_S3\_OBJECTS***

Settrue, le esportazioni degli oggetti esportati in Amazon S3 vengono mantenute. (Il valore predefinito è.) false

**--output-file** *OUTPUT\_FILE*

Specifica il percorso del file in cui salvare l'archivio di registro. Se viene fornito, i registri vengono salvati localmente. Altrimenti vengono caricati su Amazon S3 con l'URL restituito nell'output. L'impostazione predefinita è il caricamento su Amazon S3.

**--region, -r** *REGION*

Specifica l'uso Regione AWS. Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

**--start-time** *START\_TIME*

Specifica l'inizio dell'intervallo di tempo, espresso nel formato ISO 8601 (ad `YYYY-MM-DDThh:mm:ssZ` esempio `2021-01-01T20:00:00Z`). Sono inclusi gli eventi di registro con un timestamp uguale a questo orario o successivo a tale ora. Se non specificato, l'impostazione predefinita è l'ora di creazione del cluster.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster export-image-logs --bucket image-v3-bucket --image-id ami-1234abcd5678efgh
{
 "url": "https://image-v3-bucket..."
}
```

**pcluster get-cluster-log-events**

Recupera gli eventi associati a un flusso di log.

```
pcluster get-cluster-log-events [-h]
 --cluster-name CLUSTER_NAME
 --log-stream-name LOG_STREAM_NAME
 [--debug]
 [--end-time END_TIME]
 [--limit LIMIT]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
 [--start-from-head START_FROM_HEAD]
 [--start-time START_TIME]
```

## Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster get-cluster-log-events`.

### **--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

### **--log-stream-name *LOG\_STREAM\_NAME***

Specifica il nome del flusso di log. È possibile utilizzare il `list-cluster-log-streams` comando per recuperare un flusso di log associato a uno o più eventi.

### **--debug**

Abilita la registrazione del debug.

### **--end-time *END\_TIME***

Specifica la fine dell'intervallo di tempo, espresso nel formato ISO 8601 (YYYY-MM-DDThh:mm:ssZad esempio). 2021-01-01T20:00:00Z Gli eventi con un timestamp uguale o successivo a tale orario non sono inclusi.

### **--limit *LIMIT***

Specifica il numero massimo di eventi di registro restituiti. Se non viene specificato un valore, il numero massimo di eventi di registro è il numero massimo di eventi di registro che può contenere una dimensione di risposta di 1 MB, fino a 10.000 eventi di registro.

### **--next-token *NEXT\_TOKEN***

Token per il set di risultati successivo.

### **--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

### **--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

### **--start-from-head *START\_FROM\_HEAD***

Se il valore è `true`, vengono restituiti per primi i primi eventi di registro. Se il valore è `false`, gli eventi di registro più recenti vengono restituiti per primi. (Il valore predefinito è.) `false`

**--start-time *START\_TIME***

Specificate l'inizio dell'intervallo di tempo, espresso nel formato ISO 8601 (YYYY-MM-DDThh:mm:ssZad esempio). 2021-01-01T20:00:00Z Sono inclusi gli eventi con un timestamp uguale a questo orario o successivo a tale orario.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster get-cluster-log-events \
 -c cluster-v3 \
 -r us-east-1 \
 --log-stream-name ip-198-51-100-44.i-1234567890abcdef0.clustermgtd \
 --limit 3
{
 "nextToken": "f/36966906399261933213029082268132291405859205452101451780/s",
 "prevToken": "b/36966906399239632467830551644990755687586557090595471362/s",
 "events": [
 {
 "message": "2022-07-12 19:16:53,379 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Performing node maintenance actions",
 "timestamp": "2022-07-12T19:16:53.379Z"
 },
 {
 "message": "2022-07-12 19:16:53,380 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Following nodes are currently in replacement: (x0) []",
 "timestamp": "2022-07-12T19:16:53.380Z"
 },
 {
 "message": "2022-07-12 19:16:53,380 -
[slurm_plugin.clustermgtd:_terminate_orphaned_instances] - INFO - Checking for
orphaned instance",
 "timestamp": "2022-07-12T19:16:53.380Z"
 }
]
}
```

**pcluster get-cluster-stack-events**

Recupera gli eventi associati allo stack per il cluster specificato.

 Note

A partire dalla versione 3.6.0, AWS ParallelCluster utilizza gli stack annidati per creare le risorse associate alle code e alle risorse di calcolo. L'GetClusterStackEventsAPI e il `pcluster get-cluster-stack-events` comando restituiscono solo gli eventi dello stack principale del cluster. È possibile visualizzare gli eventi dello stack del cluster, inclusi quelli relativi alle code e alle risorse di calcolo, nella console. CloudFormation

```
pcluster get-cluster-stack-events [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

**-h, --help**

Mostra il testo di aiuto per `pcluster get-cluster-stack-events`.

**--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

**--debug**

Abilita la registrazione del debug.

**--next-token *NEXT\_TOKEN***

Token per il set di risultati successivo.

**--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

## Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster get-cluster-stack-events \
 -n cluster-v3 \
 -r us-east-1 \
 --query "events[0]"
{
 "eventId": "1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "physicalResourceId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "resourceStatus": "CREATE_COMPLETE",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "stackName": "cluster-v3",
 "logicalResourceId": "cluster-v3",
 "resourceType": "AWS::CloudFormation::Stack",
 "timestamp": "2022-07-12T18:29:12.140Z"
}
```

## pcluster get-image-log-events

Recupera gli eventi di registro associati alla creazione di un'immagine.

```
pcluster get-image-log-events [-h]
 --image-id IMAGE_ID
 --log-stream-name LOG_STREAM_NAME
 [--debug]
 [--end-time END_TIME]
 [--limit LIMIT]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
 [--start-from-head START_FROM_HEAD]
 [--start-time START_TIME]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster get-image-log-events`.

### --image-id, -i **IMAGE\_ID**

Specifica l'ID dell'immagine.



**--log-stream-name *LOG\_STREAM\_NAME***

Specifica il nome del flusso di log. È possibile utilizzare il `list-image-log-streams` comando per recuperare un flusso di log associato a uno o più eventi.

**--debug**

Abilita la registrazione del debug.

**--end-time *END\_TIME***

Specifica la fine dell'intervallo di tempo, espresso nel formato ISO 8601 (YYYY-MM-DDThh:mm:ssZad esempio). 2021-01-01T20:00:00Z Gli eventi con un timestamp uguale o successivo a tale orario non sono inclusi.

**--limit *LIMIT***

Specifica il numero massimo di eventi di registro restituiti. Se non viene specificato un valore, il numero massimo di eventi di registro è il numero massimo di eventi di registro che può contenere una dimensione di risposta di 1 MB, fino a 10.000 eventi di registro.

**--next-token *NEXT\_TOKEN***

Token per il set di risultati successivo.

**--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

**--start-from-head *START\_FROM\_HEAD***

Se il valore è `true`, vengono restituiti per primi i primi eventi di registro. Se il valore è `false`, gli eventi di registro più recenti vengono restituiti per primi. (Il valore predefinito è.) `false`

**--start-time *START\_TIME***

Specificate l'inizio dell'intervallo di tempo, espresso nel formato ISO 8601 (YYYY-MM-DDThh:mm:ssZad esempio). 2021-01-01T20:00:00Z Sono inclusi gli eventi con un timestamp uguale o successivo a tale orario.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster get-image-log-events --image-id custom-alinux2-image --region us-east-1 --
log-stream-name 3.1.2/1 --limit 3
{
 "nextToken": "f/36778317771100849897800729464621464113270312017760944178/s",
 "prevToken": "b/36778317766952911290874033560295820514557716777648586800/s",
 "events": [
 {
 "message": "ExecuteBash: FINISHED EXECUTION",
 "timestamp": "2022-04-05T22:13:26.633Z"
 },
 {
 "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-1234abcd-56ef-78gh-90ij-abcd1234efgh/3.1.2/1",
 "timestamp": "2022-04-05T22:13:26.741Z"
 },
 {
 "message": "TOE has completed execution successfully",
 "timestamp": "2022-04-05T22:13:26.819Z"
 }
]
}
```

## pcluster get-image-stack-events

Recupera gli eventi associati allo stack per la build dell'immagine specificata.

```
pcluster get-image-stack-events [-h]
 --image-id IMAGE_ID
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster get-image-stack-events`.

### --image-id, -i **IMAGE\_ID**

Specifica l'ID dell'immagine.

**--debug**

Abilita la registrazione del debug.

**--next-token** *NEXT\_TOKEN*

Token per il set di risultati successivo.

**--query** *QUERY*

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r** *REGION*

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di AWS\_DEFAULT\_REGION ambiente, l'regionimpostazione nella [default] sezione del ~/.aws/config file o il --region parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster get-image-stack-events --image-id custom-alinux2-image --region us-east-1 --
query "events[0]"
{
 "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
 "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "Resource creation Initiated",
 "resourceProperties": "{\"InfrastructureConfigurationArn\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\",\\\"ImageRecipeArn
\\\":\\\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-custom-alinux2-image/3.1.2\\\",\\\"DistributionConfigurationArn
\\\":\\\"arn:aws:imagebuilder:us-east-1:123456789012:distribution-
configuration/parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\",
\\\"EnhancedImageMetadataEnabled\\\":\\\"false\\\",\\\"Tags\\\":{\\\"parallelcluster:image_name\\\":
\\\"custom-alinux2-image\\\",\\\"parallelcluster:image_id\\\":\\\"custom-alinux2-image\\\"}}\",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/custom-alinux2-
image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "stackName": "custom-alinux2-image",
 "logicalResourceId": "ParallelClusterImage",
 "resourceType": "AWS::ImageBuilder::Image",
 "timestamp": "2022-04-05T21:39:24.725Z"
}
```

## pcluster list-clusters

Recupera l'elenco dei cluster esistenti.

```
pcluster list-clusters [-h]
 [--cluster-status {CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
 DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
 UPDATE_COMPLETE,UPDATE_FAILED}]
 [{CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
 DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
 UPDATE_COMPLETE,UPDATE_FAILED} ...]]
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster list-clusters`.

**--cluster-status {CREATE\_IN\_PROGRESS, CREATE\_FAILED, CREATE\_COMPLETE, DELETE\_IN\_PROGRESS, DELETE\_FAILED, UPDATE\_IN\_PROGRESS, UPDATE\_COMPLETE, UPDATE\_FAILED} [{CREATE\_IN\_PROGRESS, CREATE\_FAILED, CREATE\_COMPLETE, DELETE\_IN\_PROGRESS, DELETE\_FAILED, UPDATE\_IN\_PROGRESS, UPDATE\_COMPLETE, UPDATE\_FAILED} ...]**

Specifica l'elenco degli stati del cluster per cui filtrare. (Il valore predefinito è.) `all`

### --debug

Abilita la registrazione del debug.

### --next-token *NEXT\_TOKEN*

Token per il set di risultati successivo.

### --query *QUERY*

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster list-clusters
{
 "clusters": [
 {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_COMPLETE"
 }
]
}
```

**pcluster list-cluster-log-streams**

Recupera l'elenco dei flussi di log associati a un cluster.

```
pcluster list-cluster-log-streams [-h]
 --cluster-name CLUSTER_NAME
 [--filters FILTERS [FILTERS ...]]
 [--next-token NEXT_TOKEN] [--debug]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

**-h, --help**

Mostra il testo di aiuto per `pcluster list-cluster-log-streams`.

**--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

**--debug**

Abilita la registrazione del debug.

**--filters *FILTERS* [*FILTERS* ...]**

Specifica i filtri per i flussi di log. Formato: `Name=a,Values=1 Name=b,Values=2,3` I filtri supportati sono:

`private-dns-name`

Specifica la forma abbreviata del nome DNS privato dell'istanza (ad `ip-10-0-0-101` esempio).

`node-type`

Specifica il tipo di nodo, l'unico valore accettato per questo filtro è. `HeadNode`

**--next-token *NEXT\_TOKEN***

Token per il set di risultati successivo.

**--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster list-cluster-log-streams \
 -n cluster-v3 \
 -r us-east-1 \
 --query 'logStreams[*].logStreamName'
[
 "ip-172-31-58-205.i-1234567890abcdef0.cfn-init",
 "ip-172-31-58-205.i-1234567890abcdef0.chef-client",
```

```
"ip-172-31-58-205.i-1234567890abcdef0.cloud-init",
"ip-172-31-58-205.i-1234567890abcdef0.clustermgtd",
"ip-172-31-58-205.i-1234567890abcdef0.slurmctld",
"ip-172-31-58-205.i-1234567890abcdef0.supervisord",
"ip-172-31-58-205.i-1234567890abcdef0.system-messages"
]
```

## pcluster list-images

Recupera l'elenco delle immagini personalizzate esistenti.

```
pcluster list-images [-h]
 --image-status {AVAILABLE,PENDING,FAILED}
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster list-images`.

### --image-status {AVAILABLE,PENDING,FAILED}

Filtra le immagini restituite in base allo stato fornito.

### --debug

Abilita la registrazione del debug.

### --next-token *NEXT\_TOKEN*

Token per il set di risultati successivo.

### --query *QUERY*

Specifica l' JMESPath interrogazione da eseguire sull'output.

### --region, -r *REGION*

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

## Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster list-images --image-status AVAILABLE
{
 "images": [
 {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "BUILD_COMPLETE",
 "ec2AmiInfo": {
 "amiId": "ami-1234abcd5678efgh"
 },
 "region": "us-east-1",
 "version": "3.1.2"
 }
]
}
```

## pcluster list-image-log-streams

Recupera l'elenco dei flussi di log associati a un'immagine.

```
pcluster list-image-log-streams [-h]
 --image-id IMAGE_ID
 [--next-token NEXT_TOKEN] [--debug]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster list-image-log-streams`.

### --image-id, -i **IMAGE\_ID**

Specifica l'ID dell'immagine.

### --debug

Abilita la registrazione del debug.

### --next-token **NEXT\_TOKEN**

Token per il set di risultati successivo.



**--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di AWS\_DEFAULT\_REGION ambiente, l'regionimpostazione nella [default] sezione del ~/.aws/config file o il --region parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster list-image-log-streams --image-id custom-alinux2-image --region us-east-1 --
query 'logStreams[*].logStreamName'
[
 "3.0.0/1",
 "3.1.2/1"
]
```

**pcluster list-official-images**

Descrivi ufficiale AWS ParallelCluster AMIs.

```
pcluster list-official-images [-h]
 [--architecture ARCHITECTURE]
 [--debug]
 [--os OS]
 [--query QUERY]
 [--region REGION]
```

Argomenti denominati

**-h, --help**

Mostra il testo di aiuto per `pcluster list-official-images`.

**--architecture *ARCHITECTURE***

Specifica l'architettura da utilizzare per filtrare i risultati. Se questo parametro non è specificato, vengono restituite tutte le architetture.

**--debug**

Abilita la registrazione del debug.

**--os *OS***

Specifica il sistema operativo da utilizzare per filtrare i risultati. Se questo parametro non è specificato, vengono restituiti tutti i sistemi operativi.

**--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando l'impostazione [Region](#) nel file di configurazione dell'immagine, la variabile di AWS\_DEFAULT\_REGION ambiente, l'regionimpostazione nella [default] sezione del ~/.aws/config file o il --region parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.2:

```
$ pcluster list-official-images
{
 "images": [
 {
 "amiId": "ami-015cf4eb4e0d6306b2",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-2204-lts-hvm-x86_64-2022022615052022-02-26T15-08-34.759Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-036f23237ce49d25b",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-x86_64-2022022615052022-02-26T15-08-17.558Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-09e5327e694d89ef4",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-2204-lts-hvm-arm64-2022022615052022-02-26T15-08-45.736Z",
 "version": "3.1.2",

```

```

 "architecture": "arm64"
 },
 {
 "amiId": "ami-0b9b0874c35f626ae",
 "os": "alinux2",
 "name": "aws-parallelcluster-3.1.2-amzn2-hvm-x86_64-202202261505
2022-02-26T15-08-31.311Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-0d0de4f95f56374bc",
 "os": "alinux2",
 "name": "aws-parallelcluster-3.1.2-amzn2-hvm-arm64-202202261505
2022-02-26T15-08-46.088Z",
 "version": "3.1.2",
 "architecture": "arm64"
 },
 {
 "amiId": "ami-0ebf7bc54b8740dc6",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.293Z",
 "version": "3.1.2",
 "architecture": "arm64"
 }
]
}

```

## pcluster ssh

Esegue un ssh comando con il nome utente e l'indirizzo IP del cluster precompilati. Gli argomenti arbitrari vengono aggiunti alla fine della riga di comando. ssh

```

pcluster ssh [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--region REGION]

```

## Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster ssh`.

### **--cluster-name, -n *CLUSTER\_NAME***

Specifica il nome del cluster cui connettersi.

### **--debug**

Abilita la registrazione dei debug.

### **--dryrun *DRYRUN***

When `true`, stampa la riga di comando che verrebbe eseguita ed esce. (Il valore predefinito è.)  
`false`

### **--region, -r *REGION***

Specifica l'uso. Regione AWS Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio:

```
$ pcluster ssh --cluster-name mycluster -i ~/.ssh/id_rsa
```

Esegue un `ssh` comando con il nome utente e l'indirizzo IP del cluster precompilato:

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

## **pcluster update-cluster**

Aggiorna un cluster esistente in modo che corrisponda alle impostazioni di un file di configurazione specificato.

### Note

L'aggiornamento del cluster ha esito positivo solo se tutti i nodi del cluster hanno applicato correttamente l'aggiornamento. Se l'aggiornamento non riesce, consulta la [Vedere](#)

[clusterStatus](#) avviene `UPDATE_FAILED` tramite `pcluster describe-cluster comando` sezione per una guida alla risoluzione dei problemi.

```
pcluster update-cluster [-h]
 --cluster-configuration CLUSTER_CONFIGURATION
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--force-update FORCE_UPDATE]
 [--query QUERY]
 [--region REGION]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```

Argomenti denominati

#### **-h, --help**

Mostra il testo di aiuto per `pcluster update-cluster`.

#### **--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

Specifica il file di configurazione del cluster YAML.

#### **--cluster-name, -n *CLUSTER\_NAME***

Specificate il nome del cluster.

#### **--debug**

Abilita la registrazione del debug.

#### **--dryrun *DRYRUN***

Quando `true`, esegue la convalida senza aggiornare il cluster e creare risorse. Può essere utilizzato per convalidare la configurazione dell'immagine e aggiornare i requisiti. (Il valore predefinito è.) `false`

#### **--force-update *FORCE\_UPDATE***

Quando `true`, forza l'aggiornamento ignorando gli errori di convalida dell'aggiornamento. (Il valore predefinito è.) `false`

**--query *QUERY***

Specifica l' JMESPath interrogazione da eseguire sull'output.

**--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando l'[Region](#) impostazione nel file di configurazione del cluster, la variabile di AWS\_DEFAULT\_REGION ambiente, l'[region](#) impostazione nella [default] sezione del ~/.aws/config file o il --region parametro.

**--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

Identifica uno o più validatori di configurazione da sopprimere.

Formato: (|) ALL type: [A-Za-z0-9]+

**--validation-failure-level *{INFO,WARNING,ERROR}***

Specifica il livello di errori di convalida segnalati per l'aggiornamento.

Esempio di utilizzo AWS ParallelCluster della versione 3.1.4:

```
$ pcluster update-cluster -c cluster-config.yaml -n cluster-v3 -r us-east-1
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "UPDATE_IN_PROGRESS"
 },
 "changeSet": [
 {
 "parameter": "HeadNode.Iam.S3Access",
 "requestedValue": {
 "BucketName": "amzn-s3-demo-bucket1",
 "KeyName": "output",
 "EnableWriteAccess": false
 }
 },
 {
```

```

 "parameter": "HeadNode.Iam.S3Access",
 "currentValue": {
 "BucketName": "amzn-s3-demo-bucket2",
 "KeyName": "logs",
 "EnableWriteAccess": true
 }
 }
]
}

```

## pcluster update-compute-fleet

Aggiorna lo stato della flotta di elaborazione del cluster.

```

pcluster update-compute-fleet [-h]
 --cluster-name CLUSTER_NAME
 --status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

 [--debug]
 [--query QUERY]
 [--region REGION]

```

Argomenti denominati

### -h, --help

Mostra il testo di aiuto per `pcluster update-compute-fleet`.

### --cluster-name, -n *CLUSTER\_NAME*

Specificate il nome del cluster.

### --status {START\_REQUESTED,STOP\_REQUESTED,ENABLED,DISABLED}

Specifica lo stato applicato alla flotta di elaborazione del cluster. Gli stati `START_REQUESTED` e `STOP_REQUESTED` corrispondono allo Slurm scheduler mentre gli stati `ENABLED` e `DISABLED` corrispondono allo scheduler AWS Batch.

### --debug

Abilita la registrazione del debug.

### --query *QUERY*

Specifica l' JMESPath interrogazione da eseguire sull'output.

## **--region, -r *REGION***

Specifica l'uso Regione AWS . Regione AWS Deve essere specificato utilizzando la variabile di `AWS_DEFAULT_REGION` ambiente, l'`region` impostazione nella `[default]` sezione del `~/.aws/config` file o il `--region` parametro.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster update-compute-fleet -n cluster-v3 --status STOP_REQUESTED
{
 "status": "STOP_REQUESTED",
 "lastStatusUpdateTime": "2022-07-12T20:19:47.653Z"
}
```

## **pcluster version**

Visualizza la versione di. AWS ParallelCluster

```
pcluster version [-h] [--debug]
```

Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster version`.

### **--debug**

Abilita la registrazione dei debug.

Esempio di utilizzo della AWS ParallelCluster versione 3.1.4:

```
$ pcluster version
{
 "version": "3.1.4"
}
```

## **pcluster3-config-converter**

Legge un file di configurazione della AWS ParallelCluster versione 2 e scrive un file di configurazione della AWS ParallelCluster versione 3.



```
pcluster3-config-converter [-h]
 [-t CLUSTER_TEMPLATE]
 [-c CONFIG_FILE]
 [--force-convert]
 [-o OUTPUT_FILE]
```

## Argomenti denominati

### **-h, --help**

Mostra il testo di aiuto per `pcluster3-config-converter`.

### **-t *CLUSTER\_TEMPLATE*, --cluster-template *CLUSTER\_TEMPLATE***

Specifica la [\[cluster\]sezione](#) del file di configurazione da convertire. Se non viene specificato, lo script cercherà il parametro [cluster-template](#) nella [\[global\]sezione](#) o lo cercherà. `[cluster default]`

### **-c *CONFIG\_FILE*, --config-file *CONFIG\_FILE***

Specifica il file di configurazione della AWS ParallelCluster versione 2 da leggere.

### **--force-convert**

Abilita una conversione anche se una o più impostazioni non sono supportate e non sono consigliate.

### **-o *OUTPUT\_FILE*, --output-file *OUTPUT\_FILE***

Specifica il file di configurazione della AWS ParallelCluster versione 3 da scrivere. Se questo parametro non è specificato, la configurazione viene scritta su stdout.

#### Note

Il `pcluster3-config-converter` comando è stato aggiunto nella AWS ParallelCluster versione 3.0.1.

## File di configurazione

AWS ParallelCluster utilizza i file YAML 1.1 per i parametri di configurazione.

## Argomenti

- [File di configurazione del cluster](#)
- [Crea file di configurazione delle immagini](#)

## File di configurazione del cluster

AWS ParallelCluster la versione 3 utilizza file di configurazione separati per controllare la definizione dell'infrastruttura del cluster e la definizione di infrastruttura personalizzata. AMIs Tutti i file di configurazione utilizzano file YAML 1.1. Le informazioni dettagliate per ciascuno di questi file di configurazione sono riportate di seguito. Per alcuni esempi di configurazioni, vedi [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example\\_configs](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs).

Questi oggetti vengono utilizzati per la configurazione del cluster AWS ParallelCluster versione 3.

## Argomenti

- [Proprietà del file di configurazione del cluster](#)
- [Sezione Imds](#)
- [Sezione Image](#)
- [Sezione HeadNode](#)
- [Sezione Scheduling](#)
- [Sezione SharedStorage](#)
- [Sezione Iam](#)
- [Sezione LoginNodes](#)
- [Sezione Monitoring](#)
- [Sezione Tags](#)
- [Sezione AdditionalPackages](#)
- [Sezione DirectoryService](#)
- [Sezione DeploymentSettings](#)

## Proprietà del file di configurazione del cluster

Region(Facoltativo,String)

Specifica il Regione AWS per il cluster. Ad esempio, us-east-2.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### CustomS3Bucket(Facoltativo,**String**)

Specifica il nome di un bucket Amazon S3 creato nel AWS tuo account per archiviare le risorse utilizzate dai cluster, come il file di configurazione del cluster, e per esportare i log. AWS ParallelCluster mantiene un bucket Amazon S3 in ogni AWS regione in cui crei i cluster. Per impostazione predefinita, questi bucket Amazon S3 sono denominati `parallelcluster-hash-v1-D0-NOT-DELETE`

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito. Se si forza l'aggiornamento, il nuovo valore verrà ignorato e verrà utilizzato il vecchio valore.

### AdditionalResources(Facoltativo,) **String**

Definisce un AWS CloudFormation modello aggiuntivo da avviare insieme al cluster. Questo modello aggiuntivo viene utilizzato per creare risorse esterne al cluster ma che fanno parte del ciclo di vita del cluster.

Il valore deve essere un URL HTTPS verso un modello pubblico, con tutti i parametri forniti.

Non c'è alcun valore predefinito.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Sezione **Imds**

(Facoltativo) Specifica la configurazione del servizio globale di metadati dell'istanza (IMDS).

```
Imds:
ImdsSupport: string
```

### Proprietà **Imds**

#### ImdsSupport(Facoltativo,) **String**

Specifica quali versioni IMDS sono supportate nei nodi del cluster. I valori supportati sono `v1.0` e `v2.0`. Il valore predefinito è `v2.0`.

Se `ImdsSupport` è impostato su `v1.0`, entrambi `IMDSv2` e `IMDSv1` sono supportati.

Se `ImdsSupport` è impostato su `v2.0`, solo IMDSv2 è supportato.

Per ulteriori informazioni, consulta [Use IMDSv2](#) in the Amazon EC2 User Guide for Linux instances.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

#### Note

A partire da AWS ParallelCluster 3.7.0, il `ImdsSupport` valore predefinito è `v2.0`. Ti consigliamo di impostare `v2.0` e `ImdsSupport` sostituire IMDSv1 con IMDSv2 nelle tue azioni personalizzate le chiamate.

Il supporto per [Imds/ImdsSupport](#) viene aggiunto con la AWS ParallelCluster versione 3.3.0.

## Sezione **Image**

#### Note

Le versioni non supportate della versione ufficiale AMIs distribuita da AWS ParallelCluster verranno rese non disponibili dopo 18 mesi di inattività. Queste vecchie immagini contengono software obsoleto e non possono ricevere assistenza in caso di problemi. Consigliamo vivamente di passare all'ultima versione supportata.

(Obbligatorio) Definisce il sistema operativo per il cluster.

Image:

Os: *string*

CustomAmi: *string*

## Proprietà **Image**

`Os`(Obbligatorio,String)

Specifica il sistema operativo da utilizzare per il cluster. I valori supportati sono `alinux2`, `alinux2023`, `ubuntu2404`, `ubuntu2204`, `rhel8`, `rocky8`, `rhel9`, `rocky9`.

**Note**

RedHat Enterprise Linux 8.7 (rhel8) viene aggiunto a partire dalla AWS ParallelCluster versione 3.6.0.

Se si configura il cluster per l'utilizzo di `rhel`, il costo su richiesta per qualsiasi tipo di istanza è maggiore rispetto a quando si configura il cluster per l'utilizzo di altri sistemi operativi supportati. Per ulteriori informazioni sui prezzi, consulta la sezione Prezzi [On-Demand e In che modo viene offerto e prezzato Red Hat Enterprise Linux on Amazon EC2?](#).

RedHat Enterprise Linux 9 (rhel9) viene aggiunto a partire dalla AWS ParallelCluster versione 3.9.0.

Tutte le regioni AWS commerciali supportano tutti i seguenti sistemi operativi.

| Partizione ( )Regioni AWS                                        | <b>alinux2</b> | <b>ubuntu224</b> | <b>ubuntu244</b> | <b>rhel8</b> | <b>rhel9</b> | <b>alinux2023</b> |
|------------------------------------------------------------------|----------------|------------------|------------------|--------------|--------------|-------------------|
| Commerciale (tutto ciò Regioni AWS non espressamente menzionato) | True           | True             | True             | True         | True         | True              |
| AWS GovCloud (Stati Uniti orientali) (us-gov-east-1 )            | True           | True             | True             | True         | True         | True              |
| AWS GovCloud (Stati Uniti occidentali) (us-gov-west-1)           | True           | True             | True             | True         | True         | True              |
| Cina (Pechino) (cn-north-1 )                                     | True           | True             | True             | True         | True         | True              |
| Cina (Ningxia) (cn-northwest-1 )                                 | True           | True             | True             | True         | True         | True              |

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

**Note**

AWS ParallelCluster 3.8.0 supporta Rocky Linux 8, ma Rocky Linux 8 preconfigurato AMIs (per architetture x86 e ARM) non è disponibile. AWS ParallelCluster 3.8.0 supporta la creazione di cluster con Rocky Linux 8 utilizzando sistemi personalizzati. AMIs Per ulteriori informazioni, fare riferimento a [Considerazioni sul sistema operativo](#) AWS ParallelCluster 3.9.0 supporta Rocky Linux 9, ma Rocky Linux 9 preconfigurato AMIs (per architetture x86 e ARM) non è disponibile. AWS ParallelCluster 3.9.0 supporta la creazione di cluster con Rocky Linux 9 utilizzando sistemi personalizzati. AMIs [Per ulteriori informazioni, fare riferimento a Considerazioni sul sistema operativo.](#)

### CustomAmi(Facoltativo,**String**)

Speciifica l'ID di un'AMI personalizzata da utilizzare per i nodi principale e di calcolo anziché l'AMI predefinito. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

Se l'AMI personalizzata richiede autorizzazioni aggiuntive per il suo avvio, tali autorizzazioni devono essere aggiunte sia alle politiche utente che a quelle del nodo principale.

Ad esempio, se a un'AMI personalizzata è associata un'istantanea crittografata, sono necessarie le seguenti politiche aggiuntive sia nelle politiche utente che in quelle del nodo principale:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:us-east-1:111122223333:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

```

]
 }
}

```

Per creare un'AMI personalizzata RedHat Enterprise Linux, è necessario configurare il sistema operativo per l'installazione dei pacchetti forniti dai repository RHUI (AWS): `rhel-<version>-baseos-rhui-rpms`, `rhel-<version>-appstream-rhui-rpms` e `codeready-builder-for-rhel-<version>-rhui-rpms`. Inoltre, i repository sull'AMI personalizzata devono contenere `kernel-devel` pacchetti nella stessa versione della versione del kernel in esecuzione. `kernel`.

Limitazioni note:

- Solo RHEL 8.2 e versioni successive supportano Lustre FSx .
- La versione 4.18.0-425.3.1.el8 del kernel RHEL 8.7 non supporta Lustre. FSx
- Solo RHEL 8.4 e versioni successive supportano EFA.
- AL23 non supporta NICE DCV, in quanto non include un ambiente desktop grafico, necessario per eseguire NICE DCV. Per ulteriori informazioni, consultate la documentazione ufficiale di [NICE DCV](#).

Per risolvere i problemi relativi agli avvisi di convalida AMI personalizzati, consulta. [Risoluzione dei problemi relativi alle AMI personalizzate](#)

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## Sezione **HeadNode**

(Obbligatorio) Specifica la configurazione per il nodo principale.

HeadNode:

InstanceType: *string*

Networking:

SubnetId: *string*

ElasticIp: *string/boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

```
Proxy:
 HttpProxyAddress: string
DisableSimultaneousMultithreading: boolean
Ssh:
 KeyName: string
 AllowedIps: string
LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 DeleteOnTermination: boolean
 EphemeralVolume:
 MountDir: string
SharedStorageType: string
Dcv:
 Enabled: boolean
 Port: integer
 AllowedIps: string
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
```



```

Args:
 - string
Iam:
 InstanceRole: string
 InstanceProfile: string
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
Imds:
 Secured: boolean
Image:
 CustomAmi: string

```

## Proprietà **HeadNode**

### InstanceType(Obbligatorio,String)

Specifica il tipo di istanza per il nodo principale.

Specifica il tipo di istanza Amazon EC2 utilizzato per il nodo principale. L'architettura del tipo di istanza deve essere la stessa dell'architettura utilizzata per l'impostazione AWS Batch [InstanceType](#) o Slurm [InstanceType](#).

#### Note

AWS ParallelCluster non supporta i seguenti tipi di istanza per l'HeadNode impostazione.

- hpc6id

Se si definisce un tipo di istanza p4d o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete, è necessario impostare [ElasticIp](#) questa opzione per fornire l'accesso pubblico. true AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. In questo caso, si consiglia di utilizzare un [gateway NAT](#) per fornire l'accesso pubblico ai nodi di calcolo del cluster. Per ulteriori informazioni, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`DisableSimultaneousMultithreading(Facoltativo,)` Boolean

Se `true`, disabilita l'hyper-threading sul nodo principale. Il valore predefinito è `false`.

Non tutti i tipi di istanza possono disabilitare l'hyper-threading. Per un elenco dei tipi di istanza che supportano la disabilitazione dell'hyperthreading, consulta [Core e thread della CPU per ogni core di CPU per tipo di istanza nella Guida per l'utente](#) di Amazon EC2.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`SharedStorageType(Facoltativo,)` String

Specifica il tipo di archiviazione utilizzato per i dati condivisi internamente. I dati condivisi internamente includono i dati AWS ParallelCluster utilizzati per gestire il cluster e [Sezione SharedStorage](#) quelli predefiniti condivisi, `/home` se non specificati nella directory di montaggio, per montare un volume di filesystem condiviso. Per ulteriori dettagli sui dati condivisi interni, fare riferimento a [AWS ParallelCluster elenchi interni](#)

Se `Ebs`, questo è il tipo di storage predefinito, il nodo principale esporterà porzioni del suo volume principale come directory condivise per nodi di calcolo e nodi di accesso tramite NFS.

Se `Efs`, ParallelCluster creerà un filesystem EFS da utilizzare per dati interni condivisi e `/home`

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

#### Note

Quando il cluster è scalabile orizzontalmente, il tipo di storage EBS può presentare problemi di prestazioni poiché il nodo principale condivide i dati dal volume root con i nodi di calcolo utilizzando le esportazioni NFS. Utilizzando EFS, è possibile evitare le esportazioni NFS man mano che il cluster si espande orizzontalmente ed evitare i rallentamenti delle prestazioni ad esse associati. Si consiglia di scegliere EBS per massimizzare il read/write potenziale di file di piccole dimensioni e il processo di installazione. Scegli EFS per la scalabilità.

## Networking

(Obbligatorio) Definisce la configurazione di rete per il nodo principale.

```
Networking:
 SubnetId: string
 ElasticIp: string/boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 Proxy:
 HttpProxyAddress: string
```

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Proprietà **Networking**

SubnetId(Obbligatorio,String)

Specifica l'ID di una sottorete esistente in cui effettuare il provisioning del nodo principale.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

ElasticIp(Facoltativo,) String

Crea o assegna un indirizzo IP elastico al nodo principale. I valori supportati sono truefalse, o l'ID di un indirizzo IP elastico esistente. Il valore predefinito è false.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

SecurityGroups(Facoltativo,[String])

Elenco degli ID dei gruppi di sicurezza Amazon VPC da utilizzare per il nodo principale. Questi sostituiscono i gruppi di sicurezza AWS ParallelCluster creati se questa proprietà non è inclusa.

Verifica che i gruppi di sicurezza siano configurati correttamente per i tuoi [SharedStorage](#) sistemi.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## AdditionalSecurityGroups(Facoltativo,[String])

Elenco di ID di gruppo di sicurezza Amazon VPC aggiuntivi da utilizzare per il nodo principale.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Proxy(Facoltativo)

Specifica le impostazioni proxy per il nodo principale.

Proxy:

HttpProxyAddress:

*string*

## HttpProxyAddress(Facoltativo,String)

Definisce un server proxy HTTP o HTTPS, in genere `https://x.x.x.x:8080`.

Non c'è alcun valore predefinito.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Ssh

(Facoltativo) Definisce la configurazione per l'accesso SSH al nodo principale.

Ssh:

KeyName: *string*

AllowedIps: *string*

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Proprietà Ssh

### KeyName(Facoltativo,String)

Assegna un nome a una coppia di chiavi Amazon EC2 esistente per abilitare l'accesso SSH al nodo principale.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## AllowedIps(Facoltativo,) String

Specifica l'intervallo IP in formato CIDR o un ID dell'elenco di prefissi per le connessioni SSH al nodo principale. Il valore predefinito è `0.0.0.0/0`.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## LocalStorage

(Facoltativo) Definisce la configurazione di archiviazione locale per il nodo principale.

```
LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 DeleteOnTermination: boolean
 EphemeralVolume:
 MountDir: string
```

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà LocalStorage

### RootVolume(Obbligatorio)

Specifica il volume di archiviazione principale per il nodo principale.

```
RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 DeleteOnTermination: boolean
```

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Size(Facoltativo,**Integer**)

Specifica la dimensione del volume radice del nodo principale in gibibyte (GiB). La dimensione predefinita proviene dall'AMI. L'utilizzo di una dimensione diversa richiede il supporto dell'AMIfgrowroot.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## Encrypted(Facoltativo,Boolean)

Specifica se il volume principale è crittografato. Il valore predefinito è `true`.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## VolumeType(Facoltativo,**String**)

Specifica il tipo di [volume Amazon EBS](#). I valori supportati sono `gp2`, `gp3`, `io1`, `io2`, `sc1st1`, e `standard`. Il valore predefinito è `gp3`.

Per ulteriori informazioni, consulta [Tipi di volumi di Amazon EBS](#) nella Guida per l'utente di Amazon EC2.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## Iops(Facoltativo,Integer)

Definisce il numero di IOPS e `io1` `io2` il `gp3` tipo di volumi.

Il valore predefinito, i valori supportati e il `volume_size` rapporto `volume_iops` di rapporto variano in base a `VolumeType` e `Size`.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

`VolumeType = io1`

Valore predefinito Iops = 100

Valori supportati Iops = 100—64000 †

SizeRapporto massimo Iops = 50 IOPS per GiB. 5000 IOPS richiedono almeno 100 GiB.  
Size

VolumeType = io2

Valore predefinito = 100 Iops

Valori supportati Iops = 100—64000 (256000 per i volumi io2 Block Express) †

SizeRapporto massimo Iops = 500 IOPS per GiB. 5000 IOPS richiedono almeno 10 GiB.  
Size

VolumeType = gp3

Valore predefinito = 3000 Iops

Valori supportati Iops = 3000-16000

SizeRapporto massimo Iops = 500 IOPS per GiB. 5000 IOPS richiedono almeno 10 GiB.  
Size

† Il numero massimo di IOPS è garantito solo sulle [istanze basate sul sistema Nitro dotate](#) di più di 32.000 IOPS. Altre istanze garantiscono fino a 32.000 IOPS. io1I volumi più vecchi potrebbero non raggiungere le massime prestazioni a meno che non si [modifichi](#) il volume. io2 I volumi Block Express supportano Iops valori fino a 256000 sui tipi di R5b istanza. Per ulteriori informazioni, consulta i [volumi io2 Block Express](#) nella Guida per l'utente di Amazon EC2.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

Throughput(Facoltativo,**Integer**)

Definisce la velocità effettiva per i tipi di gp3 volume, in MiB/s. Questa impostazione è valida solo quando è. VolumeType gp3 Il valore predefinito è 125. Valori supportati: 125-1000 MiB/s

Il rapporto tra a non Iops può essere superiore Throughput a 0,25. Il throughput massimo di 1000 MiB/s richiede che l'Iopsimpostazione sia almeno 4000.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

DeleteOnTermination(Facoltativo,**Boolean**)

Specifica se il volume root deve essere eliminato quando il nodo principale viene terminato. Il valore predefinito è true.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## EphemeralVolume(Facoltativo)

Specifica i dettagli per ogni volume di istanza. Per ulteriori informazioni, consulta [Instance Store Volumes](#) nella Amazon EC2 User Guide.

EphemeralVolume:  
MountDir: *string*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## MountDir(Facoltativo,**String**)

Specifica la directory di montaggio per il volume dell'archivio delle istanze. Il valore predefinito è /scratch.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Dcv

(Facoltativo) Definisce le impostazioni di configurazione per il server Amazon DCV in esecuzione sul nodo principale.

Per ulteriori informazioni, consulta [Connect ai nodi principali e di accesso tramite Amazon DCV](#).

Dcv:  
Enabled: *boolean*  
Port: *integer*  
AllowedIps: *string*

### Important

Per impostazione predefinita, la porta Amazon DCV configurata da AWS ParallelCluster è aperta a tutti IPv4 gli indirizzi. Tuttavia, puoi connetterti a una porta Amazon DCV solo se disponi dell'URL per la sessione Amazon DCV e connetterti alla sessione Amazon DCV entro 30 secondi dal momento in cui l'URL viene restituito. `pcluster dcv-connect` Utilizza l'AllowedIps impostazione per limitare ulteriormente l'accesso alla porta Amazon DCV



con un intervallo IP in formato CIDR e usa l'`Port` impostazione per impostare una porta non standard.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà `Dcv`

`Enabled`(**Boolean** Obbligatorio,)

Specifica se Amazon DCV è abilitato sul nodo principale. Il valore predefinito è `false`.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Note

Amazon DCV genera automaticamente un certificato autofirmato utilizzato per proteggere il traffico tra il client Amazon DCV e il server Amazon DCV in esecuzione sul nodo principale. Per configurare il proprio certificato, consulta [Certificato HTTPS Amazon DCV](#).

## `Port`(Facoltativo,) Integer

Specifica la porta per Amazon DCV. Il valore predefinito è 8443.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`AllowedIps`(Facoltativo, consigliato) String

Specifica l'intervallo IP in formato CIDR per le connessioni ad Amazon DCV. Questa impostazione viene utilizzata solo quando si crea il gruppo di sicurezza. AWS ParallelCluster Il valore predefinito è `0.0.0.0/0` che consente l'accesso da qualsiasi indirizzo internet.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## CustomActions

(Facoltativo) Specifica gli script personalizzati da eseguire sul nodo principale.

```

CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string

```

## Proprietà **CustomActions**

### OnNodeStart(Facoltativo)

Specifica un singolo script o una sequenza di script da eseguire sul nodo principale prima che venga avviata qualsiasi azione di bootstrap di distribuzione del nodo. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

### Sequence(Facoltativo)

Elenco degli script da eseguire. AWS ParallelCluster esegue gli script nello stesso ordine in cui sono elencati nel file di configurazione, a partire dal primo.

### Script(Obbligatorio,String)

Specifica il file da utilizzare. Il percorso del file può iniziare con `https://` o `s3://`.

Args(Facoltativo,[String])

Elenco di argomenti da passare allo script.

Script(Obbligatorio,String)

Specifica il file da utilizzare per un singolo script. Il percorso del file può iniziare con `https:// os3://`.

Args(Facoltativo,[String])

Elenco di argomenti da passare al singolo script.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

OnNodeConfigured(Facoltativo)

Specifica un singolo script o una sequenza di script da eseguire sul nodo principale dopo il completamento delle azioni di bootstrap del nodo. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

Sequence(Facoltativo)

Specifica l'elenco degli script da eseguire.

Script(Obbligatorio,) String

Specifica il file da utilizzare. Il percorso del file può iniziare con `https:// os3://`.

Args(Facoltativo,[String])

Elenco di argomenti da passare allo script.

Script(Obbligatorio,String)

Specifica il file da utilizzare per un singolo script. Il percorso del file può iniziare con `https:// os3://`.

Args(Facoltativo,[String])

Elenco di argomenti da passare al singolo script.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## OnNodeUpdated(Facoltativo)

Specifica un singolo script o una sequenza di script da eseguire sul nodo principale dopo il completamento delle azioni di aggiornamento del nodo. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

### Sequence(Facoltativo)

Specifica l'elenco degli script da eseguire.

Script(Obbligatorio,) String

Specifica il file da utilizzare. Il percorso del file può iniziare con `https://os3://`.

Args(Facoltativo,[String])

Elenco di argomenti da passare allo script.

Script(Obbligatorio,String)

Specifica il file da utilizzare per il singolo script. Il percorso del file può iniziare con `https://os3://`.

Args(Facoltativo,[String])

Elenco di argomenti da passare al singolo script.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### Note

OnNodeUpdated viene aggiunto a partire da AWS ParallelCluster 3.4.0.

Sequence viene aggiunto a partire dalla AWS ParallelCluster versione 3.6.0. Quando si specifica Sequence, è possibile elencare più script per un'azione personalizzata. AWS ParallelCluster continua a supportare la configurazione di un'azione personalizzata con un singolo script, senza includere Sequence.

AWS ParallelCluster non supporta l'inclusione sia di un singolo script che della stessa azione personalizzata. Sequence.

## Iam

(Facoltativo) Specifica un ruolo di istanza o un profilo di istanza da utilizzare sul nodo principale per sovrascrivere il ruolo o il profilo di istanza di istanza predefinito per il cluster.

Iam:

```

InstanceRole: string
InstanceProfile: string
S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
AdditionalIamPolicies:
 - Policy: string

```

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Proprietà **Iam**

### InstanceProfile(Facoltativo,) String

Specificate un profilo di istanza per sostituire il profilo di istanza del nodo principale predefinito. Non è possibile specificare InstanceProfile e InstanceRole. Il formato è `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

Se viene specificato, le AdditionalIamPolicies impostazioni S3Access and non possono essere specificate.

Si consiglia di specificare una o entrambe AdditionalIamPolicies le impostazioni S3Access e poiché le funzionalità aggiunte richiedono AWS ParallelCluster spesso nuove autorizzazioni.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### InstanceRole(Facoltativo,**String**)

Specifica un ruolo di istanza per sovrascrivere il ruolo di istanza del nodo principale predefinito. Non è possibile specificare InstanceProfile e InstanceRole. Il formato è `arn:Partition:iam::Account:role/RoleName`.

Se viene specificato, le AdditionalIamPolicies impostazioni S3Access and non possono essere specificate.

Si consiglia di specificare una o entrambe AdditionalIamPolicies le impostazioni S3Access e poiché le funzionalità aggiunte richiedono AWS ParallelCluster spesso nuove autorizzazioni.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## S3Access

### S3Access(Facoltativo)

Specifica un bucket. Viene utilizzato per generare politiche per concedere l'accesso specificato al bucket.

Se viene specificato, le InstanceRole impostazioni InstanceProfile and non possono essere specificate.

Si consiglia di specificare una o entrambe AdditionalIamPolicies le impostazioni S3Access e poiché le funzionalità aggiunte richiedono AWS ParallelCluster spesso nuove autorizzazioni.

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### BucketName(Obbligatorio,String)

Nome del bucket .

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### KeyName(Facoltativo,String)

La chiave per il secchio. Il valore predefinito è "\*".

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### EnableWriteAccess(Facoltativo,Boolean)

Indica se l'accesso in scrittura è abilitato per il bucket. Il valore predefinito è false.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## AdditionalIamPolicies

### AdditionalIamPolicies(Facoltativo)

Specifica un elenco di Amazon Resource Names (ARN) delle policy IAM per Amazon EC2. Questo elenco è allegato al ruolo root utilizzato per il nodo principale oltre alle autorizzazioni richieste da AWS ParallelCluster.

Il nome di una policy IAM e il relativo ARN sono diversi. I nomi non possono essere usati.

Se è specificato, le InstanceRole impostazioni InstanceProfile and non possono essere specificate.

Si consiglia di utilizzarle AdditionalIamPolicies perché AdditionalIamPolicies vengono aggiunte alle autorizzazioni richieste e InstanceRole devono includere tutte le autorizzazioni richieste. AWS ParallelCluster Le autorizzazioni richieste spesso cambiano da rilascio a rilascio man mano che vengono aggiunte caratteristiche.

Non c'è alcun valore predefinito.

AdditionalIamPolicies:

- Policy: *string*

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Policy(Facoltativo,) [String]

Elenco delle politiche IAM.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Ims

(Facoltativo) Specifica le proprietà per esempio il servizio di metadati (IMDS). Per ulteriori informazioni, consulta [Come funziona il servizio di metadati delle istanze versione 2](#) nella Guida per l'utente di Amazon EC2.

Ims:

Secured: *boolean*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà **Imds**

Secured(Facoltativo,) Boolean

If `true`, limita l'accesso all'IMDS del nodo principale (e alle credenziali del profilo di istanza) a un sottoinsieme di superutenti.

Se `false`, ogni utente nel nodo principale ha accesso all'IMDS del nodo principale.

L'accesso all'IMDS del nodo principale è consentito ai seguenti utenti:

- utente `root`
- utente amministrativo del cluster (`pc-cluster-admin` per impostazione predefinita)
- utente predefinito specifico del sistema operativo (`ec2-users` su Amazon Linux 2 e RedHat ubuntu Ubuntu 18.04).

Il valore predefinito è `true`.

Gli `default` utenti hanno la responsabilità di garantire che un cluster disponga delle autorizzazioni necessarie per interagire con AWS le risorse. Se disabiliti `default` l'accesso IMDS dell'utente, non AWS ParallelCluster puoi gestire i nodi di calcolo e smette di funzionare. Non disabilitare l'accesso IMDS `default` degli utenti.

Quando a un utente viene concesso l'accesso all'IMDS del nodo principale, può utilizzare le autorizzazioni incluse nel profilo di istanza [del nodo principale](#). Ad esempio, possono utilizzare queste autorizzazioni per avviare istanze Amazon EC2 o per leggere la password per un dominio AD che il cluster è configurato per utilizzare per l'autenticazione.

Per limitare l'accesso IMDS, AWS ParallelCluster gestisce una catena di `iptables`

Gli utenti del cluster con `sudo` accesso possono abilitare o disabilitare in modo selettivo l'accesso all'IMDS del nodo principale per altri singoli utenti, inclusi gli `default` utenti, eseguendo il comando:

```
$ sudo /opt/parallelcluster/scripts/imds/imds-access.sh --allow <USERNAME>
```

È possibile disabilitare l'accesso IMDS degli utenti con l'`--deny` opzione relativa a questo comando.



Se disabiliti inconsapevolmente l'accesso IMDS default dell'utente, puoi ripristinare l'autorizzazione utilizzando l'opzione. `--allow`

#### Note

Qualsiasi personalizzazione iptables o ip6tables regola può interferire con il meccanismo utilizzato per limitare l'accesso IMDS sul nodo principale.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Image

(Facoltativo) Definisce un'immagine personalizzata per il nodo principale.

Image:

CustomAmi: *string*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà **Image**

CustomAmi(Facoltativo,String)

Specifica l'ID di un'AMI personalizzata da utilizzare per il nodo principale anziché l'AMI predefinito. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

Se l'AMI personalizzata richiede autorizzazioni aggiuntive per il suo avvio, tali autorizzazioni devono essere aggiunte sia alle politiche utente che a quelle del nodo principale.

Ad esempio, se a un'AMI personalizzata è associata un'istantanea crittografata, sono necessarie le seguenti politiche aggiuntive sia nelle politiche utente che in quelle del nodo principale:

JSON

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

```

 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:us-east-1:111122223333:key/<AWS_KMS_KEY_ID>"
]
}

```

Per risolvere i problemi relativi agli avvisi di convalida AMI personalizzati, consulta. [Risoluzione dei problemi relativi alle AMI personalizzate](#)

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Sezione **Scheduling**

(Obbligatorio) Definisce il job scheduler utilizzato nel cluster e le istanze di calcolo gestite dal job scheduler. È possibile utilizzare lo scheduler o. Slurm AWS Batch Ciascuno supporta un diverso set di impostazioni e proprietà.

Argomenti

- [Proprietà Scheduling](#)
- [AwsBatchQueues](#)
- [SlurmQueues](#)
- [SlurmSettings](#)

### Scheduling:

Scheduler: slurm

ScalingStrategy: *string*

SlurmSettings:

MungeKeySecretArn: *string*

ScaledownIdleTime: *integer*

QueueUpdateStrategy: *string*

```
EnableMemoryBasedScheduling: boolean
CustomSlurmSettings: [dict]
CustomSlurmSettingsIncludeFile: string
Database:
 Uri: string
 UserName: string
 PasswordSecretArn: string
 DatabaseName: string
ExternalSlurmdbd: boolean
 Host: string
 Port: integer
Dns:
 DisableManagedDns: boolean
 HostedZoneId: string
 UseEc2Hostnames: boolean
SlurmQueues:
 - Name: string
 ComputeSettings:
 LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 EphemeralVolume:
 MountDir: string
 CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
 CapacityType: string
 AllocationStrategy: string
 JobExclusiveAllocation: boolean
 CustomSlurmSettings: dict
 Tags:
 - Key: string
 Value: string
 HealthChecks:
 Gpu:
 Enabled: boolean
 Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
```

```
SecurityGroups:
 - string
AdditionalSecurityGroups:
 - string
PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
Proxy:
 HttpProxyAddress: string
ComputeResources:
 - Name: string
 InstanceType: string
 Instances:
 - InstanceType: string
 MinCount: integer
 MaxCount: integer
 DynamicNodePriority: integer
 StaticNodePriority: integer
 SpotPrice: float
 DisableSimultaneousMultithreading: boolean
 SchedulableMemory: integer
 HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
 CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
 Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
 CustomSlurmSettings: dict
 Tags:
 - Key: string
 Value: string
 LaunchTemplateOverrides:
 LaunchTemplateId: string
 Version: string
CustomActions:
 OnNodeStart:
```

```

Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
Iam:
 InstanceProfile: string
 InstanceRole: string
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
Image:
 CustomAmi: string

```

```

Scheduling:
 Scheduler: awsbatch
 AwsBatchQueues:
 - Name: string
 CapacityType: string
 Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 ComputeResources: # this maps to a Batch compute environment (initially we
support only 1)
 - Name: string

```

```
InstanceTypes:
 - string
MinvCpus: integer
DesiredvCpus: integer
MaxvCpus: integer
SpotBidPercentage: float
```

## Proprietà **Scheduling**

### **Scheduler**(Obbligatorio,**String**)

Specifica il tipo di scheduler utilizzato. I valori supportati sono `slurm` e `awsbatch`.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

#### Note

`awsbatch` supporta solo il sistema `linux2` operativo e `x86_64` la piattaforma.

### **ScalingStrategy**(Facoltativo,**String**)

Consente di scegliere la modalità di scalabilità Slurm dei nodi dinamici. I valori supportati sono `all-or-nothing`, `greedy-all-or-nothing` e `best-effort`. Il valore predefinito è `all-or-nothing`.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

#### Note

La strategia di scalabilità si applica solo ai nodi che devono essere ripristinati da Slurm, non ai nodi che alla fine sono già in esecuzione.

- `all-or-nothing` Questa strategia segue rigorosamente una `all-or-nothing-approach` strategia volta a evitare le istanze inattive alla fine del processo di scalabilità. Funziona su una `all-or-nothing` base, il che significa che si espande completamente o non si adatta affatto. Tieni presente che potrebbero esserci costi aggiuntivi dovuti alle istanze lanciate temporaneamente,

quando i lavori richiedono più di 500 nodi o si estendono su più risorse di elaborazione.

Questa strategia ha il throughput più basso tra le tre possibili strategie di scalabilità. Il tempo di scalabilità dipende dal numero di lavori inviati per ogni esecuzione del programma di Slurm ripresa. Inoltre, non è possibile scalare molto oltre il limite predefinito per esecuzione degli account di RunInstances risorse, che per impostazione predefinita è di 1000 istanze. Ulteriori dettagli sono disponibili nella documentazione sulla limitazione delle [API di Amazon EC2](#)

- **greedy-all-or-nothing** Analogamente alla all-or-nothing strategia, mira a evitare le istanze inattive dopo il ridimensionamento. Questa strategia consente un sovradimensionamento temporaneo durante il processo di scalabilità per ottenere un throughput più elevato rispetto all' all-or-nothing approccio, ma prevede anche lo stesso limite di scalabilità di 1000 istanze previsto dal limite di account di risorse. RunInstances
- **best-effort** Questa strategia dà la priorità a un throughput elevato, anche se ciò significa che alcune istanze potrebbero rimanere inattive al termine del processo di scalabilità. Tenta di allocare tutti i nodi richiesti dai job, ma esiste la possibilità che non soddisfi l'intera richiesta. A differenza delle altre strategie, l'approccio che prevede il massimo impegno consente di accumulare più istanze rispetto al RunInstances limite standard, a scapito delle risorse inutilizzate durante le diverse esecuzioni dei processi di scalabilità.

Ogni strategia è progettata per soddisfare diverse esigenze di scalabilità e consente di selezionarne una che soddisfi requisiti e vincoli specifici.

## AwsBatchQueues

(Facoltativo) Le impostazioni della coda. AWS Batch È supportata una sola coda. Se [Scheduler](#) è impostata su `awsbatch`, questa sezione è obbligatoria. Per ulteriori informazioni sullo `awsbatch` scheduler, vedere [configurazione della rete](#) e [Utilizzo dello scheduler AWS Batch \(awsbatch\) con AWS ParallelCluster](#).

### AwsBatchQueues:

- Name: *string*
- CapacityType: *string*
- Networking:
  - SubnetIds:
    - *string*
  - AssignPublicIp: *boolean*
  - SecurityGroups:
    - *string*
  - AdditionalSecurityGroups:

```

- string
 ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
- Name: string
 InstanceTypes:
 - string
 MinvCpus: integer
 DesiredvCpus: integer
 MaxvCpus: integer
 SpotBidPercentage: float

```

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Proprietà **AwsBatchQueues**

### **Name**(Obbligatorio,**String**)

Il nome della AWS Batch coda.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### **CapacityType**(Facoltativo,**String**)

Il tipo di risorse di calcolo utilizzate dalla AWS Batch coda. I valori supportati sono **ONDEMAND**, **oSPOT**. **CAPACITY\_BLOCK** Il valore predefinito è **ONDEMAND**.

#### Note

Se lo **CapacityType** imposti **SPOT**, il tuo account deve contenere un ruolo **AWSServiceRoleForEC2Spot** collegato al servizio. È possibile creare questo ruolo utilizzando il seguente AWS CLI comando.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Per ulteriori informazioni, consulta il [ruolo collegato ai servizi per le richieste di istanze Spot](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.



## Networking

(Obbligatorio) Definisce la configurazione di rete per la coda. AWS Batch

```
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
```

### Proprietà **Networking**

#### **SubnetIds**(Obbligatorio,[**String**])

Specifica l'ID di una sottorete esistente in cui effettuare il provisioning della AWS Batch coda. Attualmente è supportata solo una sottorete.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

#### **AssignPublicIp**(Facoltativo,**String**)

Crea o assegna un indirizzo IP pubblico ai nodi della AWS Batch coda. I valori supportati sono true e false. L'impostazione predefinita dipende dalla sottorete specificata.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

#### **SecurityGroups**(Facoltativo,[**String**])

Elenco dei gruppi di sicurezza utilizzati dalla AWS Batch coda. Se non specifichi gruppi di sicurezza, AWS ParallelCluster crea nuovi gruppi di sicurezza.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

#### **AdditionalSecurityGroups**(Facoltativo,[**String**])

Elenco dei gruppi di sicurezza utilizzati dalla AWS Batch coda.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

## ComputeResources

(Obbligatorio) Definisce la ComputeResources configurazione per la AWS Batch coda.

```
ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
- Name: string
 InstanceTypes:
 - string
 MinvCpus: integer
 DesiredvCpus: integer
 MaxvCpus: integer
 SpotBidPercentage: float
```

### Proprietà **ComputeResources**

#### **Name**(Obbligatorio,**String**)

Il nome dell'ambiente di calcolo della AWS Batch coda.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

#### **InstanceTypes**(Obbligatorio,) [**String**]

L'array di tipi di istanze dell'ambiente di AWS Batch calcolo. Tutti i tipi di istanza devono utilizzare l'x86\_64architettura.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

#### **MinvCpus**(Facoltativo,**Integer**)

Il numero minimo di dati VCPUs che un ambiente di AWS Batch calcolo può utilizzare.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

#### **DesiredVcpus**(Facoltativo,**Integer**)

Il numero desiderato di VCPUs nell'ambiente di AWS Batch calcolo. AWS Batch regola questo valore tra MinvCpus e in MaxvCpus base alla domanda nella coda dei lavori.

Politica di aggiornamento: questa impostazione non viene analizzata durante un aggiornamento.

## MaxVcpus(Facoltativo,) Integer

Il numero massimo di VCPUs per l'ambiente di AWS Batch calcolo. Non puoi impostarlo su un valore inferiore a `DesiredVcpus`.

Politica di aggiornamento: questa impostazione non può essere ridotta durante un aggiornamento.

## SpotBidPercentage(Facoltativo,Float)

La percentuale massima del prezzo on demand per il tipo di istanza che può raggiungere il prezzo di un'istanza Spot Amazon EC2 prima del lancio delle istanze. Il valore predefinito è 100 (100%). L'intervallo supportato è 1 -100.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## SlurmQueues

(Facoltativo) Impostazioni per la Slurm coda. Se [Scheduler](#) è impostata su `slurm`, questa sezione è obbligatoria.

```
SlurmQueues:
- Name: string
 ComputeSettings:
 LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 EphemeralVolume:
 MountDir: string
 CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
 CapacityType: string
 AllocationStrategy: string
 JobExclusiveAllocation: boolean
 CustomSlurmSettings: dict
 Tags:
 - Key: string
 Value: string
 HealthChecks:
```

```
Gpu:
 Enabled: boolean
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
 Proxy:
 HttpProxyAddress: string
ComputeResources:
 - Name: string
 InstanceType: string
 Instances:
 - InstanceType: string
 MinCount: integer
 MaxCount: integer
 DynamicNodePriority: integer
 StaticNodePriority: integer
 SpotPrice: float
 DisableSimultaneousMultithreading: boolean
 SchedulableMemory: integer
 HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
 CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
 Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
 CustomSlurmSettings: dict
 Tags:
 - Key: string
```

```

 Value: string
 LaunchTemplateOverrides:
 LaunchTemplateId: string
 Version: string
 CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 Iam:
 InstanceProfile: string
 InstanceRole: string
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
 Image:
 CustomAmi: string

```

Politica di aggiornamento: per questa impostazione dei valori dell'elenco, è possibile aggiungere un nuovo valore durante un aggiornamento oppure è necessario interrompere il parco di elaborazione quando si rimuove un valore esistente.

## Proprietà **SlurmQueues**

### **Name**(Obbligatorio,**String**)

Il nome della Slurm coda.

**Note**

La dimensione del cluster può cambiare durante un aggiornamento. Per ulteriori informazioni, vedere [Dimensione e aggiornamento della capacità del cluster](#)

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

**CapacityReservationTarget****Note**

CapacityReservationTarget viene aggiunto con la AWS ParallelCluster versione 3.3.0.

CapacityReservationTarget:

CapacityReservationId: *string*

CapacityReservationResourceGroupArn: *string*

Specifica la prenotazione della capacità On-Demand per le risorse di calcolo della coda.

**CapacityReservationId**(Facoltativo,) **String**

L'ID della riserva di capacità esistente a cui destinare le risorse di calcolo della coda. L'ID può fare riferimento a un [ODCR](#) o a un [Capacity Block](#) per ML.

La prenotazione deve utilizzare la stessa piattaforma utilizzata dall'istanza. Ad esempio, se le istanze vengono eseguite su `rhel8`, la prenotazione della capacità deve essere eseguita sulla piattaforma Red Hat Enterprise Linux. Per ulteriori informazioni, consultare [Piattaforme supportate](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

**Note**

Se la includete [Instances](#) nella configurazione del cluster, dovete escludere questa `CapacityReservationId` impostazione del livello di coda dalla configurazione.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **CapacityReservationResourceGroupArn**(Facoltativo,**String**)

L'Amazon Resource Name (ARN) del gruppo di risorse che funge da gruppo di prenotazioni di capacità collegato al servizio per le risorse di calcolo della coda. AWS ParallelCluster identifica e utilizza la prenotazione di capacità più appropriata del gruppo di risorse in base alle seguenti condizioni:

- Se PlacementGroup è abilitato in [SlurmQueues/Networkingo](#) [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleziona un gruppo di risorse destinato al tipo di istanza e PlacementGroup per una risorsa di calcolo, se la risorsa di calcolo esiste.

PlacementGroupDeve avere come target uno dei tipi di istanza definiti in [ComputeResources](#)

- Se PlacementGroup non è abilitato in [SlurmQueues/Networkingo](#) [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleziona un gruppo di risorse destinato solo al tipo di istanza di una risorsa di elaborazione, se la risorsa di calcolo esiste.

Il gruppo di risorse deve avere almeno un ODCR per ogni tipo di istanza riservato in una zona di disponibilità in tutte le risorse di calcolo e le zone di disponibilità della coda. Per ulteriori informazioni, consulta [Avvio di istanze con On-Demand Capacity Reservations \(ODCR\)](#).

Per ulteriori informazioni sui requisiti di configurazione di più sottoreti, vedere/. [NetworkingSubnetIds](#)

#### Note

Nella AWS ParallelCluster versione 3.4.0 sono state aggiunte più zone di disponibilità.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## CapacityType(Facoltativo,String)

Il tipo di risorse di calcolo utilizzate dalla Slurm coda. I valori supportati sono ONDEMAND, oSPOT. CAPACITY\_BLOCK Il valore predefinito è ONDEMAND.

### Note

Se lo imposti su SPOT, CapacityType il tuo account deve avere un ruolo AWSServiceRoleForEC2Spot collegato al servizio. È possibile utilizzare il seguente AWS CLI comando per creare questo ruolo.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Per ulteriori informazioni, consulta il [ruolo collegato ai servizi per le richieste di istanze Spot](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

## AllocationStrategy(Facoltativo,) String

Specificare la strategia di allocazione per tutte le risorse di calcolo definite in. [Instances](#)

Valori validi: lowest-price | capacity-optimized | price-capacity-optimized | prioritized | capacity-optimized-prioritized

| CapacityType    | Strategie consentite                                                                                |
|-----------------|-----------------------------------------------------------------------------------------------------|
| SU RICHIESTA    | al prezzo più basso, con priorità                                                                   |
| PUNTO           | prezzo più basso, con capacità ottimizzata, price-capacity-optimized capacity-optimized-prioritized |
| BLOCCO_CAPACITÀ | Non supportato: non può essere AllocationStrategy configurato                                       |



Impostazione predefinita: `lowest-price`

### **lowest-price**

- Se `utilizziCapacityType = ONDEMAND`, Amazon EC2 Fleet utilizza il prezzo per determinare l'ordine e avvia prima le istanze con il prezzo più basso.
- Se `lo utilizziCapacityType = SPOT`, Amazon EC2 Fleet lancia istanze dal pool di istanze Spot con il prezzo più basso con capacità disponibile. Se un pool esaurisce la capacità prima di soddisfare la capacità richiesta, Amazon EC2 Fleet soddisfa la tua richiesta avviando istanze per te. In particolare, Amazon EC2 Fleet lancia istanze dal pool di istanze Spot con il prezzo più basso e con capacità disponibile. Amazon EC2 Fleet potrebbe avviare istanze Spot da diversi pool.
- Se `impostataCapacityType = CAPACITY_BLOCK`, non esistono strategie di allocazione, pertanto il `AllocationStrategy` parametro non può essere configurato.

### **capacity-optimized**

- Se `impostatoCapacityType = ONDEMAND`, `capacity-optimized` non è disponibile.
- Se hai `impostatoCapacityType = SPOT`, Amazon EC2 Fleet lancia istanze da pool di istanze Spot con una capacità ottimale per il numero di istanze da avviare.

### **price-capacity-optimized**

- Se lo hai `impostatoCapacityType = ONDEMAND`, `capacity-optimized` non è disponibile.
- Se hai `impostatoCapacityType = SPOT`, Amazon EC2 Fleet identifica i pool con la massima disponibilità di capacità per il numero di istanze in fase di avvio. Ciò implica che richiederemo istanze spot dai pool che riteniamo avere le minori possibilità di interruzione nel breve termine. La flotta Amazon EC2 richiede quindi istanze Spot dal prezzo più basso di questi pool.

### **prioritized**

- Se `impostaCapacityType = ONDEMAND`, Amazon EC2 Fleet rispetta l'ordine di priorità che AWS ParallelCluster si applica alle `LaunchTemplate` sostituzioni quando vengono specificate più sottoreti. AWS ParallelCluster ricava l'`override priority` dalla posizione della sottorete di destinazione in cui la prima sottorete ottiene la massima priorità. `SlurmQueues/Networking/SubnetIds` Le priorità vengono stabilite AWS ParallelCluster in ordine decrescente da `SlurmQueues/Networking/SubnetIds`, con la prima con la priorità più alta e l'ultimo `SubnetId` `SubnetID` con la priorità più bassa.
- Se `impostata, non è disponibileCapacityType = SPOT.prioritized`

## capacity-optimized-prioritized

- Se lo imposti `CapacityType = ONDEMAND`, `capacity-optimized-prioritized` non è disponibile.
- Se imposti `CapacityType = SPOT`, Amazon EC2 Fleet ottimizza innanzitutto la capacità e poi applica, nel migliore dei modi, l'ordine di priorità assegnato alle sostituzioni. AWS ParallelCluster LaunchTemplate Le priorità vengono stabilite AWS ParallelCluster in ordine decrescente da `SlurmQueues/Networking/SubnetIds`, con la prima con la priorità più alta e l'ultimo `SubnetId` `SubnetID` con la priorità più bassa. Tutte le sostituzioni destinate alla stessa sottorete ricevono lo stesso valore di priorità.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### Note

`AllocationStrategy` è supportato a partire dalla versione 3.3.0 AWS ParallelCluster .  
Novità nella versione 3.14.0: `prioritized` (per On-Demand) e `capacity-optimized-prioritized` (per Spot).

## `JobExclusiveAllocation`(Facoltativo,) **String**

Se impostato su `true`, il `OverSubscribe` flag Slurm della partizione è impostato su `EXCLUSIVE`.  
When `OverSubscribe = EXCLUSIVE`, i job nella partizione hanno accesso esclusivo a tutti i nodi allocati. Per ulteriori informazioni, vedete [EXCLUSIVE](#) nella Slurm documentazione.

Valori validi: `true` | `false`

Default: `false`

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### Note

`JobExclusiveAllocation` è supportato a partire dalla AWS ParallelCluster versione 3.7.0.

## **CustomSlurmSettings**(Facoltativo,) **Dict**

Definisce le impostazioni di configurazione personalizzate Slurm della partizione (coda).

Specifica un dizionario di coppie chiave-valore dei parametri di Slurm configurazione personalizzati che si applicano alle code (partizioni).

Ogni coppia chiave-valore separata, ad esempio `Param1: Value1`, viene aggiunta separatamente alla fine della riga di configurazione della partizione nel Slurm formato.  
`Param1=Value1`

È possibile specificare solo parametri di Slurm configurazione che non sono elencati in `CustomSlurmSettings`. Per informazioni sui parametri di Slurm configurazione non elencati, vedere [Non inserito nell'elenco Slurm parametri di configurazione per CustomSlurmSettings](#)

AWS ParallelCluster controlla solo se un parametro si trova in un elenco negato. AWS ParallelCluster non convalida la sintassi o la semantica dei parametri di Slurm configurazione personalizzati. È tua responsabilità convalidare i parametri di configurazione personalizzati. Slurm I parametri di Slurm configurazione personalizzati non validi possono causare errori nei Slurm daemon che possono causare errori nella creazione e nell'aggiornamento del cluster.

Per ulteriori informazioni su come specificare parametri di Slurm configurazione personalizzati con, vedere [AWS ParallelCluster Slurm personalizzazione della configurazione](#)

Per ulteriori informazioni sui parametri di Slurm configurazione, vedere [slurm.conf](#) nella documentazione. Slurm

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

### Note

`CustomSlurmSettings` è supportato a partire dalla versione 3.6.0. AWS ParallelCluster

## **Tags**(Facoltativo, [String])

Un elenco di coppie chiave-valore di tag. [ComputeResource](#) i tag sostituiscono i tag duplicati specificati in o in/. [Sezione Tags](#) SlurmQueues Tags

## **Key**(Facoltativo,) **String**

La chiave di tag:

**Value**(Facoltativo,**String**)

Il valore del tag.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

**HealthChecks**(Facoltativo)

Specificate i controlli dello stato dei nodi di calcolo su tutte le risorse di calcolo nella coda.

**Gpu**(Facoltativo)

Specificate i controlli di integrità della GPU su tutte le risorse di calcolo in una coda.

 **Note**

AWS ParallelCluster non supporta `HealthChecks/Gpu` nei nodi che utilizzano sistemi operativi `alinux2` ARM. Queste piattaforme non supportano [NVIDIA Data Center GPU Manager \(DCGM\)](#).

L'abilitazione dei controlli dello stato della GPU quando si utilizzano tipi di istanze la cui dimensione totale della memoria della GPU è superiore a 327680 MiB è sconsigliata.

**Enabled**(Facoltativo,) **Boolean**

Se AWS ParallelCluster esegue controlli dello stato della GPU sui nodi di elaborazione. Il valore predefinito è `false`.

**Gpu**comportamento del controllo dello stato di salute

- Se `Gpu/Enabled` è impostato su `true`, AWS ParallelCluster esegue controlli dello stato della GPU sulle risorse di calcolo in coda.
- Il controllo `Gpu` dello stato esegue controlli sullo stato della GPU sulle risorse di elaborazione per impedire l'invio di lavori su nodi con una GPU danneggiata.
- Se un nodo di elaborazione non supera un controllo di `Gpu` integrità, lo stato del nodo di elaborazione cambia in `DRAIN`. I nuovi lavori non iniziano su questo nodo. I lavori esistenti vengono eseguiti fino al completamento. Una volta completati tutti i processi in esecuzione, il nodo di calcolo termina se si tratta di un nodo dinamico e viene sostituito se si tratta di un nodo statico.

- La durata del controllo dello Gpu stato dipende dal tipo di istanza selezionato, dal numero di oggetti GPUs presenti nell'istanza, dalla memoria totale della GPU e dal numero di obiettivi del controllo Gpu dello stato (equivalente al numero di obiettivi GPU del processo). Ad esempio, su un p4d.24xlarge, la durata tipica è di 3 minuti.
- Se il controllo di Gpu integrità viene eseguito su un'istanza non supportata, viene chiusa e il processo viene eseguito sul nodo di elaborazione. Ad esempio, se un'istanza non dispone di una GPU o se un'istanza ha una GPU ma non è una GPU NVIDIA, il controllo dello stato viene interrotto e il processo viene eseguito sul nodo di calcolo. È GPUs supportata solo la tecnologia NVIDIA.
- Il controllo dello Gpu stato utilizza lo `dcmi` strumento per eseguire controlli di integrità su un nodo e segue i seguenti passaggi:

Quando inizia il controllo dello stato di Gpu salute in un nodo:

1. Rileva se i `nvidia-fabricmanager` servizi `nvidia-dcgm` and sono in esecuzione.
2. Se questi servizi non sono in esecuzione, il controllo Gpu sanitario li avvia.
3. Rileva se la modalità di persistenza è abilitata.
4. Se la modalità di persistenza non è abilitata, il controllo dello Gpu stato la abilita.

Al termine del controllo sanitario, il controllo Gpu sanitario riporta questi servizi e risorse allo stato iniziale.

- Se il job è assegnato a un set specifico di nodi GPUs, il controllo dello Gpu stato viene eseguito solo su quel set specifico. Altrimenti, il controllo Gpu dello stato viene eseguito su tutti i GPUs componenti del nodo.
- Se un nodo di elaborazione riceve 2 o più richieste di controllo dello stato di Gpu salute contemporaneamente, viene eseguito solo il primo controllo e gli altri vengono ignorati. Questo vale anche per i controlli sanitari del nodo di destinazione. GPUs È possibile controllare i file di registro per ulteriori informazioni su questa situazione.
- Il registro dei controlli di integrità per un nodo di calcolo specifico è disponibile nel `/var/log/parallelcluster/slurm_health_check.log` file. Il file è disponibile in Amazon CloudWatch, nel gruppo di CloudWatch log del cluster, dove puoi trovare:
  - Dettagli sull'azione eseguita dal controllo dello stato di Gpu salute, tra cui l'attivazione e la disabilitazione dei servizi e la modalità di persistenza.
  - L'identificatore GPU, l'ID seriale e l'UUID.
  - L'output del controllo dello stato di salute.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

 **Note**

HealthChecks è supportato a partire dalla AWS ParallelCluster versione 3.6.0.

## Networking

(Obbligatorio) Definisce la configurazione di rete per la Slurm coda.

```
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
 Proxy:
 HttpProxyAddress: string
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### Proprietà **Networking**

#### **SubnetIds**(Obbligatorio, **[String]**)

Le IDs sottoreti esistenti in cui si effettua il provisioning della Slurm coda.

Se si configurano i tipi di istanza in [SlurmQueues/ComputeResources/InstanceType](#), è possibile definire solo una sottorete.

Se si configurano i tipi di istanza in [SlurmQueuesComputeResources//Instances](#), è possibile definire una o più sottoreti.

Se si utilizzano più sottoreti, tutte le sottoreti definite per una coda devono trovarsi nello stesso VPC, con ciascuna sottorete in una zona di disponibilità (AZ) separata.

Ad esempio, supponiamo di definire subnet-1 e subnet-2 per la coda.

subnet-1 e subnet-2 possono essere entrambi in AZ-1.

subnet-1 può essere in AZ-1 e subnet-2 può essere in AZ-2.

Se configuri solo un tipo di istanza e desideri utilizzare più sottoreti, definisci il tipo di istanza in anziché `Instances InstanceType`

Ad esempio, `definisciComputeResources/Instances/InstanceType= instance.type` anziché `ComputeResources/InstanceType=. instance.type`

 Note

Elastic Fabric Adapter (EFA) non è supportato in diverse zone di disponibilità.

L'uso di più zone di disponibilità potrebbe causare un aumento della latenza della rete di storage e costi aggiuntivi per il trasferimento dei dati tra le AZ. Ad esempio, ciò potrebbe verificarsi quando un'istanza accede allo storage di file che si trova in una zona di disponibilità diversa. Per ulteriori informazioni, consulta [Trasferimento dati all'interno della stessa Regione AWS](#).

Aggiornamenti del cluster per passare dall'uso di una singola sottorete a più sottoreti:

- Supponiamo che la definizione di sottorete di un cluster sia definita con una singola sottorete e un AWS ParallelCluster file system managed for Lustre. FSx Quindi, non è possibile aggiornare direttamente questo cluster con una definizione di ID di sottorete aggiornata. Per effettuare l'aggiornamento del cluster, è necessario innanzitutto modificare il file system gestito con un file system esterno. Per ulteriori informazioni, consulta [Convertire lo storage AWS ParallelCluster gestito in storage esterno](#).
- Supponiamo che la definizione di sottorete di un cluster sia definita con una singola sottorete e un file system Amazon EFS esterno se non esistono destinazioni di montaggio EFS per tutte le sottoreti multiple definite da aggiungere. AZs Quindi, non è possibile aggiornare direttamente questo cluster con una definizione di ID di sottorete aggiornata. Per aggiornare il cluster o creare un cluster, è necessario innanzitutto creare tutti i target di montaggio per tutte le sottoreti multiple definite. AZs

Zone di disponibilità e prenotazioni di capacità del cluster definite in:

[CapacityReservationResourceGroupArn](#)

- Non è possibile creare un cluster se non vi è alcuna sovrapposizione tra l'insieme di tipi di istanze e zone di disponibilità coperti dal gruppo di risorse di prenotazione della capacità definito e l'insieme di tipi di istanze e zone di disponibilità definiti per la coda.
- È possibile creare un cluster se esiste una sovrapposizione parziale tra l'insieme di tipi di istanze e zone di disponibilità coperti dal gruppo di risorse di prenotazione della capacità definito e l'insieme di tipi di istanze e zone di disponibilità definiti per la coda. AWS ParallelCluster invia un messaggio di avviso sulla sovrapposizione parziale in questo caso.
- Per ulteriori informazioni, consulta [Avvio di istanze con On-Demand Capacity Reservations \(ODCR\)](#).

 Note

Nella AWS ParallelCluster versione 3.4.0 sono state aggiunte più zone di disponibilità.

 Warning

Questo avviso si applica a tutte le AWS ParallelCluster versioni 3.x.y precedenti alla versione 3.3.1. AWS ParallelCluster la modifica di questo parametro non influisce sulla versione 3.3.1.

Per AWS ParallelCluster 3 versioni precedenti alla versione 3.3.1:

Se si modifica questo parametro e si aggiorna un cluster, viene creato un nuovo file system gestito FSx per Lustre ed eliminato il file system Managed FSx for Lustre esistente senza preservare i dati esistenti. Ciò si traduce in una perdita di dati. Prima di procedere, assicurati di eseguire il backup dei dati dal file system esistente di FSx for Lustre se desideri conservare i dati. Per ulteriori informazioni, vedete [Lavorare con i backup nella Guida FSx](#) per l'utente di for Lustre.

Se viene aggiunto un nuovo valore di sottorete, [Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento](#).

Se viene rimosso un valore di sottorete, [Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento](#).



## **AssignPublicIp**(Facoltativo,**String**)

Crea o assegna un indirizzo IP pubblico ai nodi della Slurm coda. I valori supportati sono `true` e `false`. La sottorete specificata determina il valore predefinito. Una sottorete con IPs impostazione pubblica predefinita per l'assegnazione di indirizzi IP pubblici.

Se si definisce un tipo di `hpc6id` istanza `p4d` o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete, è necessario impostare [HeadNode/Networking](#)/su [ElasticIp](#) per fornire l'accesso `true` pubblico. AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. In questo caso, si consiglia di utilizzare un [gateway NAT](#) per fornire l'accesso pubblico ai nodi di calcolo del cluster. In questo caso, imposta su `AssignPublicIp`. `false` Per ulteriori informazioni sugli indirizzi IP, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Amazon EC2 User Guide for Linux Instances.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## **SecurityGroups**(Facoltativo,) [**String**]

Un elenco di gruppi di sicurezza da utilizzare per la Slurm coda. Se non viene specificato alcun gruppo di sicurezza, AWS ParallelCluster crea gruppi di sicurezza per te.

Verifica che i gruppi di sicurezza siano configurati correttamente per i tuoi [SharedStorage](#) sistemi.

### Warning

Questo avviso si applica a tutti e 3. **x. y** AWS ParallelCluster versioni precedenti alla versione 3.3.0. AWS ParallelCluster la modifica di questo parametro non influisce sulla versione 3.3.0.

Per AWS ParallelCluster 3 versioni precedenti alla versione 3.3.0:

Se si modifica questo parametro e si aggiorna un cluster, viene creato un nuovo file system gestito FSx per Lustre ed eliminato il file system Managed FSx for Lustre esistente senza preservare i dati esistenti. Ciò si traduce in una perdita di dati. Assicuratevi di eseguire il backup dei dati dal file system esistente di FSx for Lustre se desiderate conservare i dati. Per ulteriori informazioni, vedere [Utilizzo dei backup](#) nella Guida FSx per l'utente di for Lustre.

**⚠ Warning**

Se abiliti [Efa](#) per le tue istanze di calcolo, assicurati che le istanze abilitate per EFA siano membri di un gruppo di sicurezza che autorizza tutto il traffico in entrata e in uscita a sé stante.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**AdditionalSecurityGroups([String]Facoltativo,)**

Un elenco di gruppi di sicurezza aggiuntivi da utilizzare per la Slurm coda.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**PlacementGroup(Facoltativo)**

Specificate le impostazioni del gruppo di posizionamento per la Slurm coda.

PlacementGroup:  
Enabled: *boolean*  
Id: *string*  
Name: *string*

Politica di aggiornamento: tutti i nodi di calcolo devono essere interrotti per l'eliminazione di un gruppo di posizionamento gestito. La flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

**Enabled(Facoltativo,) Boolean**

Indica se viene utilizzato un gruppo di posizionamento per la Slurm coda. Il valore predefinito è false.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

**Id(Facoltativo,String)**

L'ID del gruppo di posizionamento per un gruppo di posizionamento del cluster esistente utilizzato dalla Slurm coda. Assicuratevi di fornire l'ID del gruppo di collocamento e non il nome.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **Name**(Facoltativo, **String**)

Il nome del gruppo di posizionamento per un gruppo di posizionamento del cluster esistente utilizzato dalla Slurm coda. Assicuratevi di fornire il nome del gruppo di posizionamento e non l'ID.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

#### Note

- Se `PlacementGroup/Enabled` è impostato su `true`, senza un `Name` o `Id` definito, a ciascuna risorsa di calcolo viene assegnato un proprio gruppo di posizionamento gestito, a meno che [ComputeResources/Networking](#)/non [PlacementGroup](#) sia definito per sostituire questa impostazione.
- A partire dalla AWS ParallelCluster versione 3.3.0, [SlurmQueues/NetworkingPlacementGroup/Name](#) è stato aggiunto come alternativa preferita a [SlurmQueues///Networking.PlacementGroupId](#)

[PlacementGroup/Id](#) e [PlacementGroup/Name](#) sono equivalenti. Puoi usare uno dei due.

Se includi sia [PlacementGroup/Id](#) che [PlacementGroup/Name](#), AWS ParallelCluster fallisce. Puoi scegliere solo l'uno o l'altro.

Non è necessario aggiornare il cluster per utilizzare [PlacementGroup/Name](#).

- Quando si utilizza una prenotazione in blocco di capacità, non è necessario impostare un vincolo relativo al gruppo di collocamento poiché potrebbero verificarsi errori di capacità insufficienti dovuti a vincoli di posizionamento esterni alla prenotazione, anche se la prenotazione di capacità ha una capacità residua.

### **Proxy**(Facoltativo)

Specifica le impostazioni proxy per la Slurm coda.

Proxy:

HttpProxyAddress: *string*

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **HttpProxyAddress**(Facoltativo,) **String**

Definisce un server proxy HTTP o HTTPS per la Slurm coda. In genere, è.

`https://x.x.x.x:8080`

Non esiste un valore predefinito.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **Image**

(Facoltativo) Specificate l'immagine da usare per la Slurm coda. Per utilizzare la stessa AMI per tutti i nodi, utilizza l'CustomAmi impostazione nella Image sezione.

Image:

CustomAmi: *string*

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **ImageProprietà**

### **CustomAmi**(Facoltativo,**String**)

L'AMI da usare per la Slurm coda anziché quella predefinita AMIs. È possibile utilizzare il comando pcluster CLI per visualizzare un elenco delle impostazioni predefinite. AMIs

#### Note

L'AMI deve essere basata sullo stesso sistema operativo utilizzato dal nodo principale.

**pcluster list-official-images**

Se l'AMI personalizzata richiede autorizzazioni aggiuntive per il suo avvio, è necessario aggiungere queste autorizzazioni alla policy del nodo principale.

Ad esempio, se a un'AMI personalizzata è associata un'istanza crittografata, nelle politiche del nodo principale sono necessarie le seguenti politiche aggiuntive.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:us-east-1:111122223333:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

Per risolvere i problemi relativi agli avvisi di convalida AMI personalizzati, consulta [Risoluzione dei problemi relativi alle AMI personalizzate](#)

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

## ComputeResources

(Obbligatorio) Definisce la ComputeResources configurazione per la coda. Slurm

 Note

- Le dimensioni del cluster possono cambiare durante un aggiornamento. Per ulteriori informazioni, vedere [Dimensione e aggiornamento della capacità del cluster](#).
- È possibile aggiungere nuove risorse di elaborazione al cluster solo se distribuite in sottoreti che appartengono ai blocchi CIDR esistenti al momento della creazione del cluster.

ComputeResources:

```

- Name: string
 InstanceType: string
 Instances:
 - InstanceType: string
 MinCount: integer
 MaxCount: integer
 DynamicNodePriority: integer
 StaticNodePriority: integer
 SpotPrice: float
 DisableSimultaneousMultithreading: boolean
 SchedulableMemory: integer
 HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
 CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
 Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
 CustomSlurmSettings: dict
 Tags:
 - Key: string
 Value: string
 LaunchTemplateOverrides:
 LaunchTemplateId: string

```

Version: *string*

Politica di aggiornamento: per questa impostazione dei valori dell'elenco, è possibile aggiungere un nuovo valore durante un aggiornamento oppure è necessario interrompere il parco di elaborazione quando si rimuove un valore esistente.

## Proprietà **ComputeResources**

### **Name**(Obbligatorio,) **String**

Il nome dell'ambiente di calcolo della Slurm coda. Il nome può contenere fino a 25 caratteri.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### **InstanceType**(Obbligatorio,**String**)

Il tipo di istanza utilizzato in questa risorsa di Slurm calcolo. Tutti i tipi di istanze in un cluster devono utilizzare la stessa architettura di processore. Le istanze possono utilizzare l'arm64architettura x86\_64 or.

La configurazione del cluster deve definire una delle due [InstanceTypeistanze](#). Se entrambi sono definiti, AWS ParallelCluster fallisce.

Quando si definisceInstanceType, non è possibile definire più sottoreti. Se configuri solo un tipo di istanza e desideri utilizzare più sottoreti, definisci il tipo di istanza in anziché inInstances. InstanceType Per ulteriori informazioni, consulta [Networking/](#). [SubnetIds](#)

Se si definisce un tipo di hpc6id istanza p4d or o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete, è necessario avviare le istanze di calcolo nella sottorete privata come descritto in. [AWS ParallelCluster utilizzando due sottoreti](#) AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. Per ulteriori informazioni, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

### **Instances**(Obbligatorio)

Specifica l'elenco dei tipi di istanza per una risorsa di calcolo. Per specificare la strategia di allocazione per l'elenco dei tipi di istanze, vedere. [AllocationStrategy](#)

La configurazione del cluster deve definire [InstanceType](#) o [Instances](#). Se entrambi sono definiti, AWS ParallelCluster fallisce.

Per ulteriori informazioni, consulta [Allocazione di più tipi di istanza con Slurm](#).

**[Instances](#):**

- [InstanceType](#): *string*

**Note**

[A partire dalla AWS ParallelCluster versione 3.7.0, `EnableMemoryBasedScheduling` può essere abilitato se si configurano più tipi di istanza in `Istanze`.](#)

Per le AWS ParallelCluster versioni da 3.2.0 a 3.6. ~~x~~, [non](#)

[EnableMemoryBasedScheduling](#) può essere abilitato se configuri più tipi di istanza in `Istanze`.

Politica di aggiornamento: per questa impostazione dei valori dell'elenco, è possibile aggiungere un nuovo valore durante un aggiornamento oppure è necessario interrompere il parco di elaborazione quando si rimuove un valore esistente.

### **InstanceType**(Obbligatorio,**String**)

Il tipo di istanza da utilizzare in questa risorsa di Slurm calcolo. Tutti i tipi di istanze in un cluster devono utilizzare la stessa architettura di processore, x86\_64 oppure arm64.

I tipi di istanza elencati in [Instances](#) devono avere:

- Lo stesso numero di v oCPUs, se [DisableSimultaneousMultithreading](#) impostato su `true`, lo stesso numero di core.
- Lo stesso numero di acceleratori degli stessi produttori.
- EFA supportato, se [Efa/Enabled](#) impostato su `true`

I tipi di istanza elencati in [Instances](#) possono avere:

- Diverse quantità di memoria.

In questo caso, la memoria minima deve essere impostata come Slurm risorsa consumabile.



 Note

A partire dalla AWS ParallelCluster versione 3.7.0, `EnableMemoryBasedScheduling` può essere abilitata se si configurano più tipi di istanza in `Istanze`.

Per le AWS ParallelCluster versioni da 3.2.0 a 3.6. `x`, non `EnableMemoryBasedScheduling` può essere abilitato se configuri più tipi di istanza in `Istanze`.

- Schede di rete diverse.

In questo caso, il numero di interfacce di rete configurate per la risorsa di calcolo è definito dal tipo di istanza con il minor numero di schede di rete.

- Larghezza di banda di rete diversa.
- Dimensioni diverse dell'Instance Store.

Se si definisce un tipo di `hpc6id` istanza `p4d` o un altro tipo di istanza con più interfacce di rete o una scheda di interfaccia di rete, è necessario avviare le istanze di calcolo nella sottorete privata come descritto in [AWS ParallelCluster utilizzando due sottoreti](#) AWS public IPs può essere assegnato solo alle istanze avviate con una singola interfaccia di rete. Per ulteriori informazioni, consulta [Assegnare un IPv4 indirizzo pubblico durante il lancio dell'istanza](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

 Note

Instances è supportato a partire dalla versione 3.3.0 AWS ParallelCluster .

## MinCount(Facoltativo,Integer)

Il numero minimo di istanze utilizzate dalla risorsa di Slurm calcolo. Il valore predefinito è 0.

 Note

La dimensione del cluster può cambiare durante un aggiornamento. Per ulteriori informazioni, vedere [Dimensione e aggiornamento della capacità del cluster](#)

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

**MaxCount**(Facoltativo,**Integer**)

Il numero massimo di istanze utilizzate dalla risorsa di Slurm calcolo. Il valore predefinito è 10.

Quando si utilizza `CapacityType = CAPACITY_BLOCK`, `MaxCount` deve essere uguale `MinCount` e maggiore di 0, poiché tutte le istanze che fanno parte della prenotazione Capacity Block vengono gestite come nodi statici.

Al momento della creazione del cluster, il nodo principale attende che tutti i nodi statici siano pronti prima di segnalare l'avvenuta creazione del cluster. Tuttavia, quando si utilizza `CapacityType = CAPACITY_BLOCK`, i nodi che fanno parte delle risorse di calcolo associate ai Capacity Blocks non verranno presi in considerazione per questo controllo. Il cluster verrà creato anche se non tutti i blocchi di capacità configurati sono attivi.

 Note

Le dimensioni del cluster possono cambiare durante un aggiornamento. Per ulteriori informazioni, vedere [Dimensione e aggiornamento della capacità del cluster](#)

**DynamicNodePriority**(Facoltativo,**Integer**)

La priorità dei nodi dinamici in una risorsa di calcolo in coda. La priorità è associata al parametro di [Weight](#) configurazione del Slurm nodo per i nodi dinamici delle risorse di calcolo. Il valore predefinito è 1000.

Slurmdà priorità ai nodi con i valori più bassi `Weight` per primi.

 Warning

L'uso di molti `Weight` valori diversi in una Slurm partizione (coda) potrebbe rallentare la velocità di pianificazione dei lavori nella coda.

Nelle AWS ParallelCluster versioni precedenti alla versione 3.7.0, ai nodi statici e dinamici veniva assegnato lo stesso peso predefinito di 1. In questo caso, Slurm potrebbe dare priorità ai nodi dinamici inattivi rispetto ai nodi statici inattivi a causa dello schema di denominazione per i nodi statici e dinamici. Quando tutto il resto è uguale, Slurm pianifica i nodi in ordine alfabetico per nome.

 Note

DynamicNodePriority viene aggiunto nella AWS ParallelCluster versione 3.7.0.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### **StaticNodePriority**(Facoltativo,) **Integer**

La priorità dei nodi statici in una risorsa di calcolo in coda. La priorità è mappata al parametro di [Weight](#) configurazione del Slurm nodo per i nodi statici delle risorse di calcolo. Il valore predefinito è 1.

Slurmdà priorità ai nodi con i valori più bassi Weight per primi.

 Warning

L'uso di molti Weight valori diversi in una Slurm partizione (coda) potrebbe rallentare la velocità di pianificazione dei lavori nella coda.

 Note

StaticNodePriority è stato aggiunto nella versione 3.7.0. AWS ParallelCluster

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### **SpotPrice**(Facoltativo,) **Float**

Il prezzo massimo pagato per un'istanza Spot Amazon EC2 prima del lancio di qualsiasi istanza. Il valore predefinito è il prezzo On-Demand.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **`DisableSimultaneousMultithreading`**(Facoltativo, **Boolean**)

Se `true`, il multithreading sui nodi della Slurm coda è disabilitato. Il valore predefinito è `false`.

Non tutti i tipi di istanza possono disabilitare il multithreading. Per un elenco dei tipi di istanza che supportano la disabilitazione del multithreading, consulta Core [e thread della CPU per ogni core di CPU per tipo di istanza nella Guida per l'utente](#) di Amazon EC2.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

### **`SchedulableMemory`**(Facoltativo,) **Integer**

La quantità di memoria in MiB configurata nel Slurm parametro `RealMemory` per i nodi di calcolo di una risorsa di calcolo. Questo valore è il limite superiore per la memoria del nodo disponibile per i lavori quando [SlurmSettings/EnableMemoryBasedScheduling](#) è abilitato. Il valore predefinito è il 95 percento della memoria elencata nei [tipi di istanza Amazon EC2](#) e restituita dall'API Amazon EC2. [DescribeInstanceTypes](#) Assicurati di convertire i valori forniti in GiB in MiB.

Valori supportati: 1-`EC2Memory`

`EC2Memory` è la memoria (in MiB) elencata nei tipi di [istanza Amazon EC2](#) e restituita dall'API Amazon EC2. [DescribeInstanceTypes](#) Assicurati di convertire i valori forniti in GiB in MiB.

Questa opzione è particolarmente rilevante quando [SlurmSettings/EnableMemoryBasedScheduling](#) è abilitato. Per ulteriori informazioni, consulta [Slurm pianificazione basata sulla memoria](#).

#### Note

`SchedulableMemory` è supportata a partire dalla AWS ParallelCluster versione 3.2.0. A partire dalla versione 3.2.0, `RealMemory` per impostazione predefinita, AWS ParallelCluster configura i nodi di Slurm calcolo sul 95 percento della memoria restituita dall'API Amazon EC2. `DescribeInstanceTypes` Questa configurazione è indipendente dal valore di `EnableMemoryBasedScheduling`

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **HealthChecks**(Facoltativo)

Specificate i controlli di integrità su una risorsa di calcolo.

### **Gpu**(Facoltativo)

Specificate i controlli di integrità della GPU su una risorsa di calcolo.

#### **Enabled**(Facoltativo,) **Boolean**

Se AWS ParallelCluster esegue controlli dello stato della GPU sul calcolo di una risorsa in coda. Il valore predefinito è `false`.

#### Note

AWS ParallelCluster non supporta `HealthChecks/Gpu` nei nodi che utilizzano sistemi operativi `alinux2 ARM`. Queste piattaforme non supportano [NVIDIA Data Center GPU Manager \(DCGM\)](#).

## **Gpu**comportamento di controllo dello stato di salute

- Se `Gpu/Enabled` è impostato su `true`, AWS ParallelCluster esegue controlli di integrità della GPU su una risorsa di calcolo.
- Il controllo `Gpu` dello stato esegue controlli sullo stato di una risorsa di elaborazione per impedire l'invio di lavori su nodi con una GPU danneggiata.
- Se un nodo di elaborazione non supera un controllo di `Gpu` integrità, lo stato del nodo di calcolo cambia in `DRAIN`. I nuovi lavori non iniziano su questo nodo. I lavori esistenti vengono eseguiti fino al completamento. Una volta completati tutti i processi in esecuzione, il nodo di calcolo termina se si tratta di un nodo dinamico e viene sostituito se si tratta di un nodo statico.
- La durata del controllo dello `Gpu` stato dipende dal tipo di istanza selezionato, dal numero di oggetti GPU presenti nell'istanza e dal numero di obiettivi del controllo `Gpu` dello stato (equivalente al numero di obiettivi GPU del processo). Per un'istanza con 8 GPU, la durata tipica è inferiore a 3 minuti.
- Se il controllo `Gpu` dello stato viene eseguito su un'istanza non supportata, viene chiusa e il processo viene eseguito sul nodo di elaborazione. Ad esempio, se un'istanza non dispone

di una GPU o se un'istanza ha una GPU ma non è una GPU NVIDIA, il controllo dello stato viene interrotto e il processo viene eseguito sul nodo di calcolo. È supportata solo la tecnologia NVIDIA.

- Il controllo dello stato GPU utilizza lo `dcgm` strumento per eseguire controlli di integrità su un nodo e segue i seguenti passaggi:

Quando inizia il controllo dello stato di GPU salute in un nodo:

1. Rileva se i `nvidia-fabricmanager` servizi `nvidia-dcgm` and sono in esecuzione.
2. Se questi servizi non sono in esecuzione, il controllo GPU sanitario li avvia.
3. Rileva se la modalità di persistenza è abilitata.
4. Se la modalità di persistenza non è abilitata, il controllo dello stato GPU la abilita.

Al termine del controllo sanitario, il controllo GPU sanitario riporta questi servizi e risorse allo stato iniziale.

- Se il job è assegnato a un set specifico di nodi GPU, il controllo dello stato GPU viene eseguito solo su quel set specifico. Altrimenti, il controllo GPU dello stato viene eseguito su tutti i GPU componenti del nodo.
- Se un nodo di elaborazione riceve 2 o più richieste di controllo dello stato di GPU salute contemporaneamente, viene eseguito solo il primo controllo e gli altri vengono ignorati. Questo vale anche per i controlli sanitari rivolti al nodo. È possibile controllare i file di registro per ulteriori informazioni su questa situazione.
- Il registro dei controlli di integrità per un nodo di calcolo specifico è disponibile nel `/var/log/parallelcluster/slurm_health_check.log` file. Questo file è disponibile in Amazon CloudWatch, nel gruppo di CloudWatch log del cluster, dove puoi trovare:
  - Dettagli sull'azione eseguita dal controllo dello stato di GPU salute, tra cui l'attivazione e la disabilitazione dei servizi e la modalità di persistenza.
  - L'identificatore GPU, l'ID seriale e l'UUID.
  - L'output del controllo dello stato di salute.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

#### Note

`HealthChecks` è supportato a partire dalla AWS ParallelCluster versione 3.6.0.

## Efa(Facoltativo)

Specifica le impostazioni Elastic Fabric Adapter (EFA) per i nodi della coda. Slurm

Efa:

Enabled: *boolean*

GdrSupport: *boolean*

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **Enabled**(Facoltativo,) **Boolean**

Specifica che Elastic Fabric Adapter (EFA) è abilitato. Per visualizzare l'elenco delle istanze Amazon EC2 che supportano EFA, consulta [Tipi di istanze supportati nella Guida per l'utente di Amazon EC2 per le istanze Linux](#). Per ulteriori informazioni, consulta [Elastic Fabric Adapter](#). Ti consigliamo di utilizzare un cluster [SlurmQueues/Networking](#) per ridurre al minimo le latenze tra [PlacementGroup](#)le istanze.

Il valore predefinito è `false`.

#### Note

Elastic Fabric Adapter (EFA) non è supportato in diverse zone di disponibilità. Per ulteriori informazioni, consulta [SubnetIds](#).

#### Warning

Se stai definendo un gruppo di sicurezza personalizzato in [SecurityGroups](#), assicurati che le istanze abilitate per EFA siano membri di un gruppo di sicurezza che consente tutto il traffico in entrata e in uscita verso se stesso.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

**GdrSupport(BooleanFacoltativo,)**

(Facoltativo) A partire dalla AWS ParallelCluster versione 3.0.2, questa impostazione non ha alcun effetto. Il supporto Elastic Fabric Adapter (EFA) per GPUDirect RDMA (accesso diretto remoto alla memoria) è sempre abilitato se è supportato dal tipo di istanza per la risorsa di Slurm elaborazione e il sistema operativo.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy](#) deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

**CapacityReservationTarget**

[CapacityReservationTarget](#):

[CapacityReservationId](#): *string*

[CapacityReservationResourceGroupArn](#): *string*

Specifica la prenotazione di capacità su richiesta da utilizzare per la risorsa di calcolo.

**CapacityReservationId(Facoltativo,) String**

L'ID della riserva di capacità esistente a cui destinare le risorse di calcolo della coda. L'id può fare riferimento a un [ODCR](#) o a un [Capacity Block](#) per ML.

Quando questo parametro è specificato a livello di risorsa di calcolo, InstanceType è facoltativo e verrà recuperato automaticamente dalla prenotazione.

**CapacityReservationResourceGroupArn(Facoltativo,) String**

Indica l'Amazon Resource Name (ARN) del gruppo di risorse che funge da gruppo di prenotazioni di capacità collegato al servizio per la risorsa di calcolo. AWS ParallelCluster identifica e utilizza la prenotazione di capacità più appropriata del gruppo. Il gruppo di risorse deve avere almeno un ODCR per ogni tipo di istanza elencato per la risorsa di calcolo. Per ulteriori informazioni, consulta [Avvio di istanze con On-Demand Capacity Reservations \(ODCR\)](#).

- Se PlacementGroup è abilitato in [SlurmQueues/Networkingo SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleziona un gruppo di risorse destinato al tipo di istanza e a una risorsa PlacementGroup di calcolo, se esistente.

PlacementGroupDeve avere come target uno dei tipi di istanze definiti in.

[ComputeResources](#)



- Se PlacementGroup non è abilitato in [SlurmQueues/Networking](#) o [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleziona un gruppo di risorse destinato solo al tipo di istanza di una risorsa di calcolo, se esistente.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

 Note

CapacityReservationTarget viene aggiunto con la AWS ParallelCluster versione 3.3.0.

## Networking

[Networking](#):

[PlacementGroup](#):

[Enabled](#): *boolean*

[Name](#): *string*

Politica di aggiornamento: tutti i nodi di calcolo devono essere interrotti per l'eliminazione di un gruppo di posizionamento gestito. La flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### PlacementGroup (Facoltativo)

Specifica le impostazioni del gruppo di posizionamento per la risorsa di calcolo.

#### Enabled (Facoltativo,) Boolean

Indica se viene utilizzato un gruppo di collocamento per la risorsa di calcolo.

- Se impostato su `true`, senza una Name definizione, a quella risorsa di calcolo viene assegnato un proprio gruppo di posizionamento gestito, indipendentemente dall'impostazione [SlurmQueues/Networking/PlacementGroup](#).
- Se è impostata su `true`, con una Name definizione, a quella risorsa di calcolo viene assegnato il gruppo di posizionamento denominato, indipendentemente dalle impostazioni [SlurmQueues/Networking/PlacementGroup](#).

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **Name**(Facoltativo,**String**)

Il nome del gruppo di posizionamento per un gruppo di posizionamento del cluster esistente utilizzato per la risorsa di calcolo.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

#### Note

- Se entrambi `PlacementGroup/Enabled` e `Name` non sono impostati, i rispettivi valori predefiniti sono [PlacementGroup](#) e impostazioni [SlurmQueues/Networking/](#).
- Quando si utilizza una prenotazione in blocco di capacità, non è necessario impostare un vincolo relativo al gruppo di collocamento poiché potrebbero verificarsi errori di capacità insufficienti dovuti a vincoli di posizionamento esterni alla prenotazione, anche se la prenotazione di capacità ha una capacità residua.
- `ComputeResources/Networking/PlacementGroup` viene aggiunto con la versione 3.3.0. AWS ParallelCluster

### **CustomSlurmSettings**(Facoltativo,**Dict**)

(Facoltativo) Definisce le impostazioni di configurazione del Slurm nodo personalizzato (risorsa di calcolo).

Specifica un dizionario di coppie chiave-valore dei parametri di Slurm configurazione personalizzati che si applicano ai Slurm nodi (risorse di calcolo).

Ogni coppia chiave-valore separata, ad esempio `Param1: Value1`, viene aggiunta separatamente alla fine della riga di configurazione del Slurm nodo nel formato. `Param1=Value1`

È possibile specificare solo parametri Slurm di configurazione che non sono elencati in.

`CustomSlurmSettings` Per informazioni sui parametri di Slurm configurazione non elencati, vedere. [Non inserito nell'elenco Slurm parametri di configurazione per CustomSlurmSettings](#)

AWS ParallelCluster controlla solo se un parametro si trova in un elenco negato. AWS ParallelCluster non convalida la sintassi o la semantica dei parametri di Slurm configurazione personalizzati. È tua responsabilità convalidare i parametri di configurazione personalizzati. Slurm I parametri di Slurm configurazione personalizzati non validi possono causare errori nei Slurm daemon che possono causare errori nella creazione e nell'aggiornamento del cluster.

Per ulteriori informazioni su come specificare parametri di Slurm configurazione personalizzati con, vedere. AWS ParallelCluster [Slurm personalizzazione della configurazione](#)

Per ulteriori informazioni sui parametri di Slurm configurazione, vedere [slurm.conf](#) nella documentazione. Slurm

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

 Note

CustomSlurmSettings è supportato a partire dalla versione 3.6.0. AWS ParallelCluster

## **Tags**(Facoltativo, [String])

Un elenco di coppie chiave-valore di tag. ComputeResource i tag sostituiscono i tag duplicati specificati in o/. [Sezione Tags SlurmQueues](#) Tags

**Key**(Facoltativo,) **String**

La chiave di tag:

**Value**(Facoltativo, **String**)

Il valore del tag.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **LaunchTemplateOverrides**(Facoltativo)

 Note

LaunchTemplateOverrides viene aggiunto con la AWS ParallelCluster versione 3.15.0.

Specifica un modello di avvio per sostituire il modello di avvio predefinito AWS ParallelCluster creato per la risorsa di calcolo. Il modello di avvio deve contenere solo sostituzioni delle interfacce di rete. AWS ParallelCluster convalida il modello di avvio e impedisce la sovrascrittura di altri parametri. Per ulteriori informazioni su come utilizzare questo override, consulta. [Personalizza le interfacce di rete dei nodi di calcolo con sostituzioni dei modelli di avvio](#)

```
LaunchTemplateOverrides:
 LaunchTemplateId: string
 Version: string
```

### **LaunchTemplateId**(Obbligatorio,**String**)

ID del modello di avvio.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy](#) deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **Version**(Obbligatorio,**String**)

Numero di versione del modello di lancio.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy](#) deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **ComputeSettings**

(Obbligatorio) Definisce la ComputeSettings configurazione per la Slurm coda.

### Proprietà **ComputeSettings**

Specifica le proprietà ComputeSettings dei nodi nella Slurm coda.

```
ComputeSettings:
 LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
```

```
Iops: integer
Throughput: integer
EphemeralVolume:
MountDir: string
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **LocalStorage**(Facoltativo)

Specifica le proprietà LocalStorage dei nodi nella Slurm coda.

```
LocalStorage:
RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
EphemeralVolume:
 MountDir: string
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **RootVolume**(Facoltativo)

Specifica i dettagli del volume principale dei nodi nella Slurm coda.

```
RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **Size**(Facoltativo,) **Integer**

Specifica la dimensione del volume root in gibibyte (GiB) per i nodi della coda. Slurm La dimensione predefinita proviene dall'AMI. L'utilizzo di una dimensione diversa richiede il supporto dell'AMIngrowroot.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

## **Encrypted**(Facoltativo,**Boolean**)

Set true, il volume principale dei nodi nella Slurm coda è crittografato. Il valore predefinito è true.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

## **VolumeType**(Facoltativo,**String**)

Specifica il [tipo di volume Amazon EBS](#) dei nodi nella Slurm coda. I valori supportati sono gp2,,gp3,io1, io2sc1, st1 e. standard Il valore predefinito è gp3.

Per ulteriori informazioni, consulta [Tipi di volumi di Amazon EBS](#) nella Guida per l'utente di Amazon EC2.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

## **Iops**(Facoltativo,**Boolean**)

Definisce il numero di IOPS e io1 io2 il gp3 tipo di volumi.

Il valore predefinito, i valori supportati e il volume\_size rapporto volume\_iops di rapporto variano in base a VolumeType eSize.

### **VolumeType = io1**

Valore predefinito Iops = 100

Valori supportati Iops = 100—64000 †

volume\_sizeRapporto massimo volume\_iops = 50 IOPS per GiB. 5000 IOPS richiedono almeno 100 GiB. volume\_size

### **VolumeType = io2**

Valore predefinito = 100 Iops

Valori supportati Iops = 100—64000 (256000 per i volumi io2 Block Express) †

SizeRapporto massimo Iops = 500 IOPS per GiB. 5000 IOPS richiedono almeno 10 GiB. Size

### **VolumeType = gp3**

Valore predefinito = 3000 Iops

Valori supportati Iops = 3000—16000 †

SizeRapporto massimo Iops = 500 IOPS per GiB per volumi con IOPS superiori a 3000.

† Il numero massimo di IOPS è garantito solo sulle [istanze basate sul sistema Nitro dotate](#) anche di più di 32.000 IOPS. Altre istanze possono avere fino a 32.000 IOPS. io1I volumi precedenti potrebbero non raggiungere le massime prestazioni a meno che non si [modifichi](#) il volume. io2 I volumi Block Express supportano volume\_iops valori fino a 256000 sui tipi di R5b istanza. Per ulteriori informazioni, consulta i [volumi io2 Block Express](#) nella Guida per l'utente di Amazon EC2.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategydeve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

### **Throughput(Facoltativo,Integer)**

Definisce la velocità effettiva per i tipi di gp3 volume, in MiB/s. Questa impostazione è valida solo quando è. VolumeType gp3 Il valore predefinito è 125. Valori supportati: 125-1000 MiB/s

Il rapporto tra a non Iops può essere superiore Throughput a 0,25. Il throughput massimo di 1000 MiB/s richiede che l'Iopsimpostazione sia almeno 4000.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategydeve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

## **EphemeralVolume**(Facoltativo,**Boolean**)

Specifica le impostazioni per il volume effimero. Il volume temporaneo viene creato combinando tutti i volumi dell'Instance Store in un unico volume logico formattato con il file system. ext4 Il valore predefinito è /scratch. Se il tipo di istanza non dispone di volumi di Instance Store, non viene creato alcun volume temporaneo. Per ulteriori informazioni, consulta [Volumi Instance store](#) nella Guida per l'utente di Amazon EC2.

EphemeralVolume:  
MountDir: *string*

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **MountDir**(Facoltativo,) **String**

La directory di montaggio per il volume temporaneo per ogni nodo della coda. Slurm

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **CustomActions**

(Facoltativo) Specifica gli script personalizzati da eseguire sui nodi della coda. Slurm

```
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
```



```
Script: string
Args:
 - string
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

## **CustomActions**Proprietà

### **OnNodeStart**(Facoltativo,**String**)

Specifica una sequenza di script o un singolo script da eseguire sui nodi della Slurm coda prima che venga avviata qualsiasi azione di bootstrap di distribuzione del nodo. AWS ParallelCluster non supporta l'inclusione sia di un singolo script che della stessa azione Sequence personalizzata. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

#### **Sequence**(Facoltativo)

Elenco degli script da eseguire.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

#### **Script**(Obbligatorio,**String**)

Il file da usare. Il percorso del file può iniziare con `https://os3://`.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

#### **Args**(Facoltativo,**[String]**)

L'elenco degli argomenti da passare allo script.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

#### **Script**(Obbligatorio,**String**)

Il file da usare per un singolo script. Il percorso del file può iniziare con `https://os3://`.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

### **Args**(Facoltativo, **[String]**)

L'elenco degli argomenti da passare al singolo script.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

### **OnNodeConfigured**(Facoltativo, **String**)

Specifica una sequenza di script o un singolo script da eseguire sui nodi della Slurm coda dopo che tutte le azioni di bootstrap del nodo sono state completate. AWS ParallelCluster non supporta l'inclusione sia di un singolo script che della stessa azione Sequence personalizzata. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

### **Sequence**(Facoltativo)

Elenco degli script da eseguire.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

### **Script**(Obbligatorio, **String**)

Il file da usare. Il percorso del file può iniziare con `https://` o `os3://`.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.](#)

### **Args**(Facoltativo, **[String]**)

L'elenco degli argomenti da passare allo script.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **Script**(Obbligatorio,**String**)

Il file da usare per un singolo script. Il percorso del file può iniziare con `https://os3://`.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

### **Args**(Facoltativo,**[String]**)

Un elenco di argomenti da passare al singolo script.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

#### Note

Sequence viene aggiunto a partire dalla AWS ParallelCluster versione 3.6.0. Quando si specifica Sequence, è possibile elencare più script per un'azione personalizzata. AWS ParallelCluster continua a supportare la configurazione di un'azione personalizzata con un singolo script, senza includere Sequence. AWS ParallelCluster non supporta l'inclusione sia di un singolo script che della stessa azione personalizzata. Sequence

## **Iam**

(Facoltativo) Definisce le impostazioni IAM opzionali per la Slurm coda.

```
Iam:
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
```

```
KeyName: string
AdditionalIamPolicies:
- Policy: string
InstanceProfile: string
InstanceRole: string
```

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## IamProprietà

### InstanceProfile(Facoltativo,String)

Specificate un profilo di istanza per sostituire il ruolo o il profilo di istanza predefinito per la Slurm coda. Non è possibile specificare entrambi e. InstanceProfile InstanceRole Il formato è `arn:${Partition}:iam:${Account}:instance-profile/${InstanceProfileName}`.

Se è specificato, le AdditionalIamPolicies impostazioni S3Access and non possono essere specificate.

Si consiglia di specificare una o entrambe AdditionalIamPolicies le impostazioni S3Access e poiché le funzionalità aggiunte richiedono AWS ParallelCluster spesso nuove autorizzazioni.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

### InstanceRole(Facoltativo,String)

Specifica un ruolo di istanza per sovrascrivere il ruolo o il profilo di istanza predefinito per la Slurm coda. Non è possibile specificare entrambi e. InstanceProfile InstanceRole Il formato è `arn:${Partition}:iam:${Account}:role/${RoleName}`.

Se è specificato, le AdditionalIamPolicies impostazioni S3Access and non possono essere specificate.

Si consiglia di specificare una o entrambe AdditionalIamPolicies le impostazioni S3Access e poiché le funzionalità aggiunte richiedono AWS ParallelCluster spesso nuove autorizzazioni.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### S3Access(Facoltativo)

Specifica un bucket per la Slurm coda. Viene utilizzato per generare politiche per concedere l'accesso specificato al bucket nella coda. Slurm

Se viene specificato, le InstanceRole impostazioni InstanceProfile and non possono essere specificate.

Si consiglia di specificare una o entrambe AdditionalIamPolicies le impostazioni S3Access e poiché le funzionalità aggiunte richiedono AWS ParallelCluster spesso nuove autorizzazioni.

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**BucketName**(Obbligatorio,**String**)

Nome del bucket .

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**KeyName**(Facoltativo,**String**)

La chiave per il secchio. Il valore predefinito è \*.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**EnableWriteAccess**(Facoltativo,**Boolean**)

Indica se l'accesso in scrittura è abilitato per il bucket.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**AdditionalIamPolicies**(Facoltativo)

Specifica un elenco di Amazon Resource Names (ARN) delle policy IAM per Amazon EC2. Questo elenco è allegato al ruolo root utilizzato per la Slurm coda oltre alle autorizzazioni richieste da. AWS ParallelCluster

Il nome di una policy IAM e il relativo ARN sono diversi. I nomi non possono essere usati.

Se è specificato, le InstanceRole impostazioni InstanceProfile and non possono essere specificate.

Si consiglia di utilizzarle `AdditionalIamPolicies` perché `AdditionalIamPolicies` vengono aggiunte alle autorizzazioni richieste e `InstanceRole` devono includere tutte le autorizzazioni richieste. AWS ParallelCluster Le autorizzazioni richieste spesso cambiano da rilascio a rilascio man mano che vengono aggiunte caratteristiche.

Non esiste un valore predefinito.

`AdditionalIamPolicies`:

- `Policy`: *string*

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**Policy**(Obbligatorio, **[String]**)

Elenco delle politiche IAM.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## SlurmSettings

(Facoltativo) Definisce le impostazioni Slurm che si applicano all'intero cluster.

`SlurmSettings`:

`ScaledownIdleTime`: *integer*

`QueueUpdateStrategy`: *string*

`EnableMemoryBasedScheduling`: *boolean*

`CustomSlurmSettings`: *[dict]*

`CustomSlurmSettingsIncludeFile`: *string*

`Database`:

`Uri`: *string*

`UserName`: *string*

`PasswordSecretArn`: *string*

`ExternalSlurmdbd`:

`Host`: *string*

`Port`: *integer*

`Dns`:

`DisableManagedDns`: *boolean*

`HostedZoneId`: *string*

`UseEc2Hostnames`: *boolean*

## SlurmSettingsProprietà

### ScaledownIdleTime(Facoltativo,Integer)

Definisce il periodo di tempo (in minuti) in cui non c'è nessun lavoro e il Slurm nodo termina.

Il valore predefinito è 10.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

### MungeKeySecretArn(Facoltativo,String)

L'Amazon Resource Name (ARN) del segreto in testo semplice di Secrets AWS Manager che contiene la chiave munge con codifica base64 da utilizzare nel cluster. Slurm Questa chiave di munge verrà utilizzata per autenticare le chiamate RPC tra comandi Slurm client e Slurm demoni che agiscono come server remoti. Se non MungeKeySecretArn viene fornito, AWS ParallelCluster genererà una chiave di munge casuale per il cluster.

#### Note

MungeKeySecretArn è supportato a partire dalla AWS ParallelCluster versione 3.8.0.

#### Warning

Se MungeKeySecretArn è stato appena aggiunto a un cluster esistente, non ParallelCluster ripristinerà la chiave munge precedente in caso di rollback o di rimozione successiva di. MungeKeySecretArn Verrà invece generata una nuova chiave di munge casuale.

Se l' AWS ParallelCluster utente ha il permesso di accedere [DescribeSecreta](#) quella specifica risorsa segreta, MungeKeySecretArn viene convalidata. MungeKeySecretArn è valido se:

- Il segreto specificato esiste e
- Il segreto è in testo semplice e contiene una stringa valida con codifica in base64 e
- La chiave di munge binaria decodificata ha una dimensione compresa tra 256 e 8192 bit.

Se la policy IAM dell'utente `pcluster` non include `DescribeSecret`, non viene convalidata e `MungeKeySecretArn` viene visualizzato un messaggio di avviso. Per ulteriori informazioni, consulta [Politica AWS ParallelCluster `pcluster` utente di base](#).

Quando esegui l'aggiornamento `MungeKeySecretArn`, la flotta di elaborazione e tutti i nodi di accesso devono essere interrotti.

Se il valore segreto nell'ARN segreto viene modificato mentre l'ARN rimane lo stesso, il cluster non verrà aggiornato automaticamente con la nuova chiave munge. Per utilizzare la nuova chiave munge dell'ARN segreto, devi interrompere la flotta di calcolo e accedere ai nodi, quindi eseguire il seguente comando dal nodo principale.

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

Dopo aver eseguito il comando, puoi riattivare sia il parco di calcolo che i nodi di accesso: i nodi di calcolo e di accesso appena assegnati inizieranno automaticamente a utilizzare la nuova chiave munge.

Per generare una chiave di munge personalizzata con codifica base64, è possibile utilizzare l'utilità [mungekey distribuita con il software di munge e quindi codificarla utilizzando l'utilità](#) `base64` generalmente disponibile nel sistema operativo. In alternativa, potete usare `bash` (impostate il parametro `bs` tra 32 e 1024)

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

o Python come segue:

```
import random
import os
import base64

key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

[Politica di aggiornamento: la flotta di elaborazione e i nodi di accesso devono essere interrotti affinché questa impostazione venga modificata per un aggiornamento.](#)



## QueueUpdateStrategy(Facoltativo,String)

Specifica la strategia di sostituzione per i parametri di [SlurmQueues](#) sezione che hanno la seguente politica di aggiornamento:

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata affinché questa impostazione venga modificata per un aggiornamento.

Il QueueUpdateStrategy valore viene utilizzato solo all'avvio di un processo di aggiornamento del cluster.

Valori validi: COMPUTE\_FLEET\_STOP | DRAIN | TERMINATE

Valore predefinito: COMPUTE\_FLEET\_STOP

### DRAIN

I nodi nelle code con valori dei parametri modificati sono impostati su. DRAINING I nodi in questo stato non accettano nuovi lavori e i lavori in esecuzione continuano fino al completamento.

Dopo che un nodo diventa idle (DRAINED), un nodo viene sostituito se il nodo è statico e il nodo viene terminato se il nodo è dinamico. Gli altri nodi in altre code senza valori dei parametri modificati non vengono influenzati.

Il tempo necessario a questa strategia per sostituire tutti i nodi della coda con valori dei parametri modificati dipende dal carico di lavoro in esecuzione.

### COMPUTE\_FLEET\_STOP

Il valore predefinito del parametro. QueueUpdateStrategy Con questa impostazione, l'aggiornamento dei parametri indicati nella [SlurmQueues](#) sezione richiede l'[interruzione del parco di elaborazione](#) prima di eseguire un aggiornamento del cluster:

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

### TERMINATE

Nelle code con valori dei parametri modificati, i processi in esecuzione vengono terminati e i nodi vengono spenti immediatamente.

I nodi statici vengono sostituiti e i nodi dinamici vengono terminati.

Gli altri nodi in altre code senza valori dei parametri modificati non vengono influenzati.

Politica di aggiornamento: questa impostazione non viene analizzata durante un aggiornamento.

 Note

QueueUpdateStrategy è supportato a partire dalla AWS ParallelCluster versione 3.2.0.

## **EnableMemoryBasedScheduling**(Facoltativo,**Boolean**)

Set true, la pianificazione basata sulla memoria è abilitata in Slurm. Per ulteriori informazioni, vedere [SlurmQueues](#)/. [ComputeResourcesSchedulableMemory](#)

Il valore predefinito è false.

 Warning

L'abilitazione della pianificazione basata sulla memoria influisce sul modo in cui lo Slurm scheduler gestisce i lavori e l'allocazione dei nodi.

Per ulteriori informazioni, consulta [Slurm pianificazione basata sulla memoria](#).

 Note

EnableMemoryBasedScheduling è supportato a partire dalla versione 3.2.0. AWS ParallelCluster

 Note

A partire dalla AWS ParallelCluster versione 3.7.0, EnableMemoryBasedScheduling può essere abilitato se si configurano più tipi di istanza in Istanze.

Per le AWS ParallelCluster versioni da 3.2.0 a 3.6. **x**, non

EnableMemoryBasedScheduling può essere abilitato se configuri più tipi di istanza in Istanze.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

### **CustomSlurmSettings**(Facoltativo, **[Dict]**)

Definisce le Slurm impostazioni personalizzate che si applicano all'intero cluster.

Specifica un elenco di dizionari di Slurm configurazione di coppie chiave-valore da aggiungere alla fine del file generato. `slurm.conf` AWS ParallelCluster

Ogni dizionario nell'elenco viene visualizzato come una riga separata aggiunta al file di configurazione. Slurm È possibile specificare parametri semplici o complessi.

I parametri semplici sono costituiti da una singola coppia di key pair, come illustrato negli esempi seguenti:

```
- Param1: 100
- Param2: "SubParam1,SubParam2=SubValue2"
```

Esempio reso in Slurm configurazione:

```
Param1=100
Param2=SubParam1,SubParam2=SubValue2
```

I parametri di Slurm configurazione complessi sono costituiti da più coppie chiave-valore separate da spazi, come mostrato negli esempi seguenti:

```
- NodeName: test-nodes[1-10]
 CPUs: 4
 RealMemory: 4196
 ... # other node settings
- NodeSet: test-nodeset
 Nodes: test-nodes[1-10]
 ... # other nodeset settings
- PartitionName: test-partition
 Nodes: test-nodeset
 ... # other partition settings
```

Esempio, reso in configurazione: Slurm

```
NodeName=test-nodes[1-10] CPUs=4 RealMemory=4196 ... # other node settings
NodeSet=test-nodeset Nodes=test-nodes[1-10] ... # other nodeset settings
```

```
PartitionName=test-partition Nodes=test-nodeset ... # other partition settings
```

**Note**

SlurmI nodi personalizzati non devono contenere i -dy- modelli -st- o nei loro nomi. Questi modelli sono riservati ai nodi gestiti da AWS ParallelCluster.

Se si specificano parametri Slurm di configurazione personalizzati in `CustomSlurmSettings`, non è necessario specificare parametri di Slurm configurazione personalizzati per `CustomSlurmSettingsIncludeFile`.

È possibile specificare solo parametri Slurm di configurazione che non sono negati in `CustomSlurmSettings`. Per informazioni sui parametri di Slurm configurazione non elencati, vedere [Non inserito nell'elenco Slurm parametri di configurazione per CustomSlurmSettings](#)

AWS ParallelCluster controlla solo se un parametro si trova in un elenco negato. AWS ParallelCluster non convalida la sintassi o la semantica dei parametri di Slurm configurazione personalizzati. È tua responsabilità convalidare i parametri di configurazione personalizzati. Slurm I parametri di Slurm configurazione personalizzati non validi possono causare errori nei Slurm daemon che possono causare errori nella creazione e nell'aggiornamento del cluster.

Per ulteriori informazioni su come specificare parametri di Slurm configurazione personalizzati con, vedere [AWS ParallelCluster Slurm personalizzazione della configurazione](#)

Per ulteriori informazioni sui parametri di Slurm configurazione, vedere [slurm.conf](#) nella documentazione. Slurm

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

**Note**

`CustomSlurmSettings` è supportato a partire dalla versione 3.6.0. AWS ParallelCluster

**CustomSlurmSettingsIncludeFile(Facoltativo,) String**

Definisce le Slurm impostazioni personalizzate che si applicano all'intero cluster.

Specifica il Slurm file personalizzato costituito da parametri di Slurm configurazione personalizzati da aggiungere alla fine del `slurm.conf` file generato AWS ParallelCluster .

È necessario includere il percorso del file. Il percorso può iniziare con `https://os3://`.

Se si specificano parametri Slurm di configurazione personalizzati per `CustomSlurmSettingsIncludeFile`, non è necessario specificare parametri di Slurm configurazione personalizzati per `CustomSlurmSettings`.

 Note

SlurmI nodi personalizzati non devono contenere i `-dy-` modelli `-st-` o nei loro nomi. Questi modelli sono riservati ai nodi gestiti da AWS ParallelCluster.

È possibile specificare solo parametri Slurm di configurazione che non sono elencati in `CustomSlurmSettingsIncludeFile`. Per informazioni sui parametri di Slurm configurazione non elencati, vedere [Non inserito nell'elenco Slurm parametri di configurazione per CustomSlurmSettings](#).

AWS ParallelCluster controlla solo se un parametro si trova in un elenco negato. AWS ParallelCluster non convalida la sintassi o la semantica dei parametri di Slurm configurazione personalizzati. È tua responsabilità convalidare i parametri di configurazione personalizzati. Slurm I parametri di Slurm configurazione personalizzati non validi possono causare errori nei Slurm daemon che possono causare errori nella creazione e nell'aggiornamento del cluster.

Per ulteriori informazioni su come specificare parametri di Slurm configurazione personalizzati con, vedere [AWS ParallelCluster Slurm personalizzazione della configurazione](#).

Per ulteriori informazioni sui parametri di Slurm configurazione, vedere [slurm.conf](#) nella documentazione. Slurm

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

 Note

`CustomSlurmSettings` è supportato a partire dalla versione 3.6.0. AWS ParallelCluster

## Database

(Facoltativo) Definisce le impostazioni per abilitare l'`SlurmAccounting` nel cluster. Per ulteriori informazioni, consulta [Slurmcontabilità con AWS ParallelCluster](#).

**Database:****Uri:** *string***UserName:** *string***PasswordSecretArn:** *string*

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

**Proprietà Database****Uri**(Obbligatorio,**String**)

L'indirizzo del server di database utilizzato come backend per la Slurm contabilità. Questo URI deve essere formattato come `host:port` e non deve contenere uno schema, ad esempio. `mysql://` L'host può essere un indirizzo IP o un nome DNS risolvibile dal nodo principale. Se non viene fornita una porta, AWS ParallelCluster utilizza la porta MySQL predefinita 3306.

AWS ParallelCluster avvia il database di Slurm contabilità nel cluster e deve accedere al database.

Il database deve essere raggiungibile prima che si verifichi quanto segue:

- Viene creato un cluster.
- Slurm la contabilità è abilitata con un aggiornamento del cluster.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

**UserName**(Obbligatorio,**String**)

L'identità Slurm utilizzata per connettersi al database, scrivere registri contabili ed eseguire query. L'utente deve disporre delle autorizzazioni di lettura e scrittura sul database.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

**PasswordSecretArn**(Obbligatorio,**String**)

L'Amazon Resource Name (ARN) del Gestore dei segreti AWS segreto che contiene la password in chiaro `UserName`. Questa password viene utilizzata insieme `UserName` all'account per l'autenticazione sul server del database. Slurm

 Note

- Quando crei un segreto utilizzando la Gestione dei segreti AWS console, assicurati di selezionare «Altro tipo di segreto», seleziona il testo non crittografato e includi solo il testo della password nel segreto.
- Non è possibile utilizzare il carattere '#' nella password del database poiché Slurm non lo supporta in slurmdbd.conf.
- [Per ulteriori informazioni su come creare un segreto, consulta Create an Gestione dei segreti AWS Secret. Gestione dei segreti AWS](#)

Se l'utente ha il permesso di farlo [DescribeSecret](#), PasswordSecretArn viene convalidato. PasswordSecretArn è valido se il segreto specificato esiste. Se la policy IAM dell'utente non include DescribeSecret, PasswordSecretArn non viene convalidata e viene visualizzato un messaggio di avviso. Per ulteriori informazioni, consulta [Politica AWS ParallelCluster pcluster utente di base](#).

Quando esegui l'aggiornamento PasswordSecretArn, la flotta di elaborazione deve essere interrotta. Se il valore segreto cambia e l'ARN segreto rimane invariato, il cluster non viene aggiornato automaticamente con la nuova password del database. Per aggiornare il cluster per il nuovo valore segreto, è necessario eseguire il comando seguente dall'interno del nodo principale dopo l'arresto del parco di elaborazione.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

 Warning

Si consiglia di modificare la password del database solo quando la flotta di elaborazione viene interrotta per evitare la perdita di dati contabili.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

**DatabaseName**(Facoltativo,**String**)

Nome del database sul server di database (definito dal parametro Uri) da utilizzare per la Slurm contabilità.

Il nome del database può contenere lettere minuscole, numeri e caratteri di sottolineatura. Il nome non può superare i 64 caratteri.

Questo parametro è mappato al StorageLoc parametro di [slurmdbd.conf](#).

Se non DatabaseName viene fornito, ParallelCluster utilizzerà il nome del cluster per definire un valore per. StorageLoc

L'aggiornamento di DatabaseName è consentito, con le seguenti considerazioni:

- Se un database con un nome DatabaseName non esiste ancora sul server del database, slurmdbd lo creerà. Sarà tua responsabilità riconfigurare il nuovo database secondo necessità (ad esempio aggiungendo le entità contabili: cluster, account, utenti, associazioni, QOSs ecc.).
- Se sul server del database esiste DatabaseName già un database con un nome, slurmdbd lo utilizzerà per la funzionalità di contabilità. Slurm

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

#### Note

Database viene aggiunto a partire dalla versione 3.3.0.

## ExternalSlurmdbd

(Facoltativo) Definisce le impostazioni per abilitare l'SlurmAccounting con un server slurmdbd esterno. [Per ulteriori informazioni, vedere Slurm Accounting with. AWS ParallelCluster](#)

ExternalSlurmdbd:

Host: *string*

Port: *integer*

## Proprietà **ExternalSlurmdbd**

### **Host**(Obbligatorio,**String**)

L'indirizzo del server slurmdbd esterno per la contabilità. Slurm L'host può essere un indirizzo IP o un nome DNS risolvibile dal nodo principale.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)



## Port(Facoltativo,) Integer

La porta che il servizio slurmdbd ascolta. Il valore predefinito è 6819.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Dns

(Facoltativo) Definisce le impostazioni Slurm che si applicano all'intero cluster.

Dns:

DisableManagedDns: *boolean*

HostedZoneId: *string*

UseEc2Hostnames: *boolean*

### Proprietà Dns

#### DisableManagedDns(Facoltativo,Boolean)

Se `true`, le voci DNS per il cluster non vengono create e i nomi dei Slurm nodi non sono risolvibili.

Per impostazione predefinita, AWS ParallelCluster crea una zona ospitata sulla Route 53 in cui i nodi vengono registrati al momento dell'avvio. Il valore predefinito è `false`. Se `DisableManagedDns` è impostato su `true`, la zona ospitata non viene creata da AWS ParallelCluster.

Per informazioni su come utilizzare questa impostazione per distribuire cluster in sottoreti senza accesso a Internet, consulta [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#)

#### Warning

È necessario un sistema di risoluzione dei nomi per il corretto funzionamento del cluster. Se `DisableManagedDns` è impostato su `true`, è necessario fornire un sistema di risoluzione dei nomi. Per utilizzare il DNS predefinito di Amazon EC2, imposta su `UseEc2Hostnames true`. In alternativa, configura il tuo resolver DNS e assicurati che i nomi dei nodi siano registrati all'avvio delle istanze. Ad esempio, puoi farlo configurando [CustomActionsOnNodeStart](#)

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## HostedZoneId(Facoltativo,String)

Definisce un ID di zona ospitata Route 53 personalizzato da utilizzare per la risoluzione dei nomi DNS per il cluster. Se fornito, AWS ParallelCluster registra i nodi del cluster nella zona ospitata specificata e non crea una zona ospitata gestita.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## UseEc2Hostnames(Facoltativo,Boolean)

Se `true`, i nodi di calcolo del cluster sono configurati con il nome host EC2 predefinito. `SlurmNodeHostName` viene inoltre aggiornato con queste informazioni. Il valore predefinito è `false`.

Per informazioni su come utilizzare questa impostazione per distribuire cluster in sottoreti senza accesso a Internet, consulta [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#)

### Note

Questa nota non è rilevante a partire dalla versione 3.3.0. AWS ParallelCluster

Per le versioni AWS ParallelCluster supportate precedenti alla 3.3.0:

Quando `UseEc2Hostnames` è impostato su `true`, il file di configurazione Slurm viene impostato con gli AWS ParallelCluster prolog script and: epilog

- `prolog` viene eseguito per aggiungere informazioni `/etc/hosts` sui nodi ai nodi di calcolo quando ogni lavoro viene allocato.
- `epilog` viene eseguito per pulire i contenuti scritti da `prolog`

Per aggiungere script `prolog` o `epilog` script personalizzati, aggiungili rispettivamente alle `/opt/slurm/etc/pcluster/epilog.d/` cartelle `/opt/slurm/etc/pcluster/prolog.d/` o.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Sezione **SharedStorage**

(Facoltativo) Le impostazioni di archiviazione condivisa per il cluster.

AWS ParallelCluster [supporta l'utilizzo di volumi di storage condivisi Amazon EBS, FSx per ONTAP e FSx OpenZFS](#), i file system di storage condivisi Amazon EFS e FSx per Lustre o File Caches.

Nella SharedStorage sezione, puoi definire lo storage esterno o gestito:

- Lo storage esterno si riferisce a un volume o file system esistente che gestisci. AWS ParallelCluster non lo crea o lo elimina.
- AWS ParallelCluster lo storage gestito si riferisce a un volume o a un file system che AWS ParallelCluster è stato creato e può eliminare.

Per le [quote di archiviazione condivisa](#) e ulteriori informazioni su come configurare lo storage condiviso, vedere [Archiviazione condivisa](#) in Utilizzo AWS ParallelCluster.

#### Note

Se AWS Batch viene utilizzato come scheduler, FSx for Lustre è disponibile solo sul nodo principale del cluster.

#### SharedStorage:

- MountDir: *string*  
Name: *string*  
StorageType: Ebs  
EbsSettings:  
VolumeType: *string*  
Iops: *integer*  
Size: *integer*  
Encrypted: *boolean*  
KmsKeyId: *string*  
SnapshotId: *string*  
Throughput: *integer*  
VolumeId: *string*  
DeletionPolicy: *string*  
Raid:  
Type: *string*  
NumberOfVolumes: *integer*
- MountDir: *string*  
Name: *string*  
StorageType: Efs  
EfsSettings:  
Encrypted: *boolean*

```

 KmsKeyId: string
 EncryptionInTransit: boolean
 IamAuthorization: boolean
 PerformanceMode: string
 ThroughputMode: string
 ProvisionedThroughput: integer
 FileSystemId: string
 DeletionPolicy: string
 AccessPointId: string
- MountDir: string
 Name: string
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: integer
 DeploymentType: string
 ImportedFileChunkSize: integer
 DataCompressionType: string
 ExportPath: string
 ImportPath: string
 WeeklyMaintenanceStartTime: string
 AutomaticBackupRetentionDays: integer
 CopyTagsToBackups: boolean
 DailyAutomaticBackupStartTime: string
 PerUnitStorageThroughput: integer
 BackupId: string
 KmsKeyId: string
 FileSystemId: string
 AutoImportPolicy: string
 DriveCacheType: string
 StorageType: string
 DeletionPolicy: string
 DataRepositoryAssociations:
 - Name: string
 BatchImportMetadataOnCreate: boolean
 DataRepositoryPath: string
 FileSystemPath: string
 ImportedFileChunkSize: integer
 AutoExportPolicy: string
 AutoImportPolicy: string
- MountDir: string
 Name: string
 StorageType: FsxOntap
 FsxOntapSettings:
 VolumeId: string

```

```

- MountDir: string
 Name: string
 StorageType: FsxOpenZfs
 FsxOpenZfsSettings:
 VolumeId: string
- MountDir: string
 Name: string
 StorageType: FileCache
 FileCacheSettings:
 FileCacheId: string

```

## SharedStorage aggiornare le politiche

- Per managed/external EBS, managed EFS e managed FSx Lustre, la policy di aggiornamento è [Politica di aggiornamento: per questa impostazione dei valori dell'elenco, la flotta di elaborazione deve essere interrotta o QueueUpdateStrategy deve essere impostata per aggiungere un nuovo valore; la flotta di elaborazione deve essere interrotta quando si rimuove un valore esistente.](#)
- Per EFS, FSx Lustre, FSx ONTAP FSx OpenZfs e File Cache esterni, la politica di aggiornamento è: [Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

## Proprietà SharedStorage

### MountDir(Obbligatorio,) String

Il percorso in cui è montato lo storage condiviso.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

### Name(Obbligatorio,String)

Il nome dello storage condiviso. Questo nome viene utilizzato quando si aggiornano le impostazioni.

#### Warning

Se si specifica lo storage condiviso AWS ParallelCluster gestito e si modifica il valore diName, lo storage condiviso gestito e i dati esistenti vengono eliminati e viene creato un nuovo storage condiviso gestito. La modifica del valore di Name con un aggiornamento del cluster equivale a sostituire lo storage condiviso gestito esistente con uno nuovo.

Assicurati di eseguire il backup dei dati prima di modificarli Name se devi conservare i dati dello storage condiviso esistente.

Politica di aggiornamento: per questa impostazione dei valori dell'elenco, la flotta di elaborazione deve essere interrotta o `QueueUpdateStrategy` deve essere impostata per aggiungere un nuovo valore; la flotta di elaborazione deve essere interrotta quando si rimuove un valore esistente.

`StorageType`(Obbligatorio,String)

Il tipo di archiviazione condivisa. I valori supportati sono `EbsEfs`, `FsxLustre`, `FsxOntap`, e `FsxOpenZfs`.

Per ulteriori informazioni, consultare [FsxLustreSettings](#), [FsxOntapSettings](#) e [FsxOpenZfsSettings](#).

 Note

Se si utilizza AWS Batch come scheduler, FSx for Lustre è disponibile solo sul nodo principale del cluster.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## EbsSettings

(Facoltativo) Le impostazioni per un volume Amazon EBS.

```
EbsSettings:
VolumeType: string
Iops: integer
Size: integer
Encrypted: boolean
KmsKeyId: string
SnapshotId: string
VolumeId: string
Throughput: integer
DeletionPolicy: string
```

Raid:Type: *string*NumberOfVolumes: *integer*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà **EbsSettings**

Quando [DeletionPolicy](#) è impostato su `Delete`, un volume gestito, con i relativi dati, viene eliminato se il cluster viene eliminato o se il volume viene rimosso con un aggiornamento del cluster.

Per ulteriori informazioni, vedere [Archiviazione condivisa](#) in Uso AWS ParallelCluster.

### VolumeType(Facoltativo,String)

Specifica il tipo di [volume Amazon EBS](#). I valori supportati sono `gp2`, `gp3`, `io1`, `io2`, `sc1`, `st1`, e `standard`. Il valore predefinito è `gp3`.

Per ulteriori informazioni, consulta [Tipi di volumi di Amazon EBS](#) nella Guida per l'utente di Amazon EC2.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Iops(Facoltativo,Integer)

Definisce il numero di IOPS e `io1` `io2` il `gp3` tipo di volumi.

Il valore predefinito, i valori supportati e il `volume_size` rapporto `volume_iops` di rapporto variano in base a `VolumeType` e `Size`.

`VolumeType = io1`

Valore predefinito Iops = 100

Valori supportati Iops = 100—64000 †

`volume_size` Rapporto massimo `volume_iops` = 50 IOPS per ogni GiB. 5000 IOPS richiedono almeno 100 GiB. `volume_size`

`VolumeType = io2`

Valore predefinito = 100 Iops

Valori supportati Iops = 100—64000 (256000 per i volumi io2 Block Express) †

SizeRapporto massimo Iops = 500 IOPS per ogni GiB. 5000 IOPS richiedono almeno 10 GiB. Size

VolumeType = gp3

Valore predefinito = 3000 Iops

Valori supportati Iops = 3000-16000

SizeRapporto massimo Iops = 500 IOPS per ogni GiB. 5000 IOPS richiedono almeno 10 GiB. Size

† Il numero massimo di IOPS è garantito solo sulle [istanze basate sul sistema Nitro dotate](#) di più di 32.000 IOPS. Altre istanze garantiscono fino a 32.000 IOPS. A meno che non si [modifichi il volume](#), i io1 volumi precedenti potrebbero non raggiungere le massime prestazioni. io2 I volumi Block Express supportano volume\_iops valori fino a 256000 sui tipi di R5b istanza. Per ulteriori informazioni, consulta i [volumi io2 Block Express](#) nella Guida per l'utente di Amazon EC2.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Size(Facoltativo,**Integer**)

Specifica la dimensione del volume in gibibyte (GiB). Il valore predefinito è 35.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

Encrypted(Facoltativo,Boolean)

Specifica se il volume è crittografato. Il valore predefinito è true.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

KmsKeyId(Facoltativo,**String**)

Specifica una AWS KMS chiave personalizzata da utilizzare per la crittografia. Questa impostazione richiede che sia Encrypted impostata su. true

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.



## SnapshotId(Facoltativo,String)

Specifica l'ID dello snapshot di Amazon EBS se utilizzi uno snapshot come origine per il volume.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## VolumeId(Facoltativo,) String

Specifica l'ID del volume Amazon EBS. Quando viene specificato per un'EbsSettingsistanza, è possibile MountDir specificare anche solo il parametro.

Il volume deve essere creato nella stessa zona di disponibilità diHeadNode.

### Note

Nella AWS ParallelCluster versione 3.4.0 sono state aggiunte più zone di disponibilità.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Throughput(Facoltativo,Integer)

La velocità effettiva, in termini MiB/s di fornitura di un volume, con un massimo di 1.000 MiB/s.

Questa impostazione è valida solo quando è VolumeType gp3 L'intervallo supportato è compreso tra 125 e 1000, con un valore predefinito di 125.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## DeletionPolicy(Facoltativo,String)

Specifica se il volume deve essere mantenuto, eliminato o creato un'istantanea quando il cluster viene eliminato o il volume viene rimosso. I valori supportati sono, e. Delete Retain Snapshot Il valore predefinito è Delete.

Quando è [DeletionPolicy](#) impostato suDelete, un volume gestito, con i relativi dati, viene eliminato se il cluster viene eliminato o se il volume viene rimosso con un aggiornamento del cluster.

Per ulteriori informazioni, consulta [Archiviazione condivisa](#).

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**Note**

DeletionPolicy è supportato a partire dalla AWS ParallelCluster versione 3.2.0.

## Raid

(Facoltativo) Definisce la configurazione di un volume RAID.

Raid:

Type: *string*

NumberOfVolumes: *integer*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Proprietà **Raid**

Type(Obbligatorio,String)

Definisce il tipo di array RAID. I valori supportati sono «0" (a strisce) e «1" (con mirroring).

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

NumberOfVolumes(Facoltativo,) Integer

Definisce il numero di volumi Amazon EBS da utilizzare per creare l'array RAID. L'intervallo di valori supportato è compreso tra 2 e 5. Il valore predefinito (quando l'Raidimpostazione è definita) è 2.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## EfsSettings

(Facoltativo) Le impostazioni per un file system Amazon EFS.

EfsSettings:

Encrypted: *boolean*

KmsKeyId: *string*

EncryptionInTransit: *boolean*

```
IamAuthorization: boolean
PerformanceMode: string
ThroughputMode: string
ProvisionedThroughput: integer
FileSystemId: string
DeletionPolicy: string
AccessPointId: string
```

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà **EfsSettings**

Quando è [DeletionPolicy](#) impostato su `Delete`, un file system gestito, con i relativi dati, viene eliminato se il cluster viene eliminato o se il file system viene rimosso con un aggiornamento del cluster.

Per ulteriori informazioni, vedere [Archiviazione condivisa](#) in Uso AWS ParallelCluster.

### Encrypted(Facoltativo, Boolean)

Specifica se il file system Amazon EFS è crittografato. Il valore predefinito è `false`.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### KmsKeyId(Facoltativo, **String**)

Specifica una AWS KMS chiave personalizzata da utilizzare per la crittografia. Questa impostazione richiede che sia `Encrypted` impostata su `true`.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### EncryptionInTransit(Facoltativo, Boolean)

Se impostato su `true`, i file system Amazon EFS vengono montati utilizzando Transport Layer Security (TLS). Per impostazione predefinita, è impostato su `false`.

#### Note

Se AWS Batch viene utilizzato come scheduler, `EncryptionInTransit` non è supportato.

**Note**

EncryptionInTransit viene aggiunto a partire dalla AWS ParallelCluster versione 3.4.0.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

**IamAuthorization**(Facoltativo, **Boolean**)

IamAuthorization viene aggiunto a partire dalla AWS ParallelCluster versione 3.4.0.

Se impostato su `true`, Amazon EFS viene autenticato utilizzando l'identità IAM del sistema. Per impostazione predefinita, è impostato su `false`.

**Note**

Se IamAuthorization è impostato su `true`, anche EncryptionInTransit deve essere impostato su `true`.

**Note**

Se AWS Batch viene utilizzato come scheduler, IamAuthorization non è supportato.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

**PerformanceMode**(Facoltativo, **String**)

Specifica la modalità di prestazioni del file system Amazon EFS. I valori supportati sono `generalPurpose` e `maxIO`. Il valore predefinito è `generalPurpose`. Per ulteriori informazioni, consulta [Modalità prestazionali](#) nella Guida per l'utente di Amazon Elastic File System.

Consigliamo la Modalità performance `generalPurpose` per la maggior parte dei file system.

I file system che utilizzano la Modalità performance `maxIO` sono in grado di scalare a livelli più elevati di throughput aggregato e operazioni al secondo. Tuttavia, esiste un compromesso tra latenze leggermente superiori per la maggior parte delle operazioni sui file.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

ThroughputMode(Facoltativo,) String

Specifica la modalità di throughput del file system Amazon EFS. I valori supportati sono `bursting` e `provisioned`. Il valore predefinito è `bursting`. Quando `provisioned` viene utilizzato, `ProvisionedThroughput` deve essere specificato.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

ProvisionedThroughput(Obbligatorio quando `ThroughputMode` è `provisioned`, Integer)

Definisce il throughput assegnato (in MiB/s) of the Amazon EFS file system, measured in MiB/s. Corrisponde al [ProvisionedThroughputInMibps](#) parametro nell'Amazon EFS API Reference.

Se si utilizza questo parametro, è necessario impostare `ThroughputMode` su `provisioned`.

L'intervallo supportato è 1 -1024. Per richiedere un aumento del limite, contatta Supporto.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

FileSystemId(Facoltativo,String)

Definisce l'ID del file system Amazon EFS per un file system esistente.

Se il cluster è configurato per estendersi su più zone di disponibilità, è necessario definire una destinazione di montaggio del file system in ogni zona di disponibilità utilizzata dal cluster.

Quando viene specificato, `MountDir` può essere specificato solo. Nessun altro `EfsSettings` può essere specificato.

Se impostate questa opzione, per i file system che definite deve valere quanto segue:

- I file system dispongono di una destinazione di montaggio esistente in ciascuna delle zone di disponibilità del cluster, con traffico NFS in entrata e in uscita consentito da e verso. HeadNode ComputeNodes [Più zone di disponibilità sono configurate in Scheduling//Networking/SlurmQueues. SubnetIds](#)

Per assicurarti che il traffico sia consentito tra il cluster e il file system, puoi effettuare una delle seguenti operazioni:

- Configura i gruppi di sicurezza del target di montaggio per consentire il traffico da e verso il CIDR o l'elenco di prefissi delle sottoreti del cluster.

 Note

AWS ParallelCluster verifica che le porte siano aperte e che il CIDR o l'elenco dei prefissi sia configurato. AWS ParallelCluster non convalida il contenuto del blocco o dell'elenco di prefissi CIDR.

- Imposta gruppi di sicurezza personalizzati per i nodi del cluster utilizzando [SlurmQueues//SecurityGroupse NetworkingHeadNode/](#). [NetworkingSecurityGroups](#) I gruppi di sicurezza personalizzati devono essere configurati per consentire il traffico tra il cluster e il file system.

 Note

Se tutti i nodi del cluster utilizzano gruppi di sicurezza personalizzati, verifica AWS ParallelCluster solo che le porte siano aperte. AWS ParallelCluster non convalida che l'origine e la destinazione siano configurate correttamente.

 Warning

EFS OneZone è supportato solo se tutti i nodi di elaborazione e il nodo principale si trovano nella stessa zona di disponibilità. EFS OneZone può avere un solo target di montaggio.

 Note

Nella AWS ParallelCluster versione 3.4.0 sono state aggiunte più zone di disponibilità.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

DeletionPolicy(Facoltativo,**String**)

Specifica se il file system deve essere mantenuto o eliminato quando il file system viene rimosso dal cluster o il cluster viene eliminato. I valori supportati sono Delete e Retain. Il valore predefinito è Delete.

Quando [DeletionPolicy](#) è impostato su `Delete`, un file system gestito, con i relativi dati, viene eliminato se il cluster viene eliminato o se il file system viene rimosso con un aggiornamento del cluster.

Per ulteriori informazioni, consulta [Archiviazione condivisa](#).

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

#### Note

`DeletionPolicy` è supportato a partire dalla AWS ParallelCluster versione 3.3.0.

## AccessPointId(Facoltativo, **String**)

Se viene specificata questa opzione, `access point ID` verrà montato il punto di ingresso del filesystem definito da e non la radice del filesystem.

Per ulteriori informazioni, consulta [Archiviazione condivisa](#).

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## FsxLustreSettings

#### Note

È necessario definire se è specificato per `FsxLustreSettings` `FsxLustreStorageType`

(Facoltativo) Le impostazioni per un file system FSx for Lustre.

```
FsxLustreSettings:
StorageCapacity: integer
DeploymentType: string
ImportedFileChunkSize: integer
DataCompressionType: string
ExportPath: string
```

```
ImportPath: string
WeeklyMaintenanceStartTime: string
AutomaticBackupRetentionDays: integer
CopyTagsToBackups: boolean
DailyAutomaticBackupStartTime: string
PerUnitStorageThroughput: integer
BackupId: string # BackupId cannot coexist with some of the fields
KmsKeyId: string
FileSystemId: string # FileSystemId cannot coexist with other fields
AutoImportPolicy: string
DriveCacheType: string
StorageType: string
DeletionPolicy: string
```

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

#### Note

Se AWS Batch viene utilizzato come scheduler, FSx for Lustre è disponibile solo sul nodo principale del cluster.

## Proprietà **FsxLustreSettings**

Quando [DeletionPolicy](#) è impostato su `Delete`, un file system gestito, con i relativi dati, viene eliminato se il cluster viene eliminato o se il file system viene rimosso con un aggiornamento del cluster.

Per ulteriori informazioni, consulta [Archiviazione condivisa](#).

### StorageCapacity(Obligatorio,Integer)

Imposta la capacità di archiviazione del file system FSx for Lustre, in GiB. `StorageCapacity` è necessario se si sta creando un nuovo file system. Non include `StorageCapacity` se `FileSystemId` è specificato `BackupId` o.

- Per i `SCRATCH_2` tipi di `PERSISTENT_2` implementazione e per i tipi di implementazione, i valori validi sono 1200 GiB, 2400 GiB e incrementi di 2400 GiB. `PERSISTENT_1`
- Per il tipo di distribuzione `SCRATCH_1`, i valori validi sono 1200 GiB, 2400 GiB e incrementi di 3600 GiB.



Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

DeploymentType(Facoltativo,) String

Specifica il tipo di distribuzione del file system FSx for Lustre. I valori supportati sono SCRATCH\_1, SCRATCH\_2, PERSISTENT\_1 e PERSISTENT\_2. Il valore predefinito è SCRATCH\_2.

Scegli SCRATCH\_1 o SCRATCH\_2 distribuzione quando hai bisogno di archiviazione temporanea e di elaborazione dei dati a breve termine. Il tipo di SCRATCH\_2 implementazione prevede la crittografia in transito dei dati e una capacità di throughput burst superiore rispetto a SCRATCH\_1.

Scegli il tipo di PERSISTENT\_1 implementazione per lo storage a lungo termine e per carichi di lavoro incentrati sul throughput che non sono sensibili alla latenza. PERSISTENT\_1 supporta la crittografia dei dati in transito. È disponibile in tutte le Regioni AWS ovunque sia disponibile FSx for Lustre.

Scegli il tipo di PERSISTENT\_2 implementazione per lo storage a lungo termine e per i carichi di lavoro sensibili alla latenza che richiedono i massimi livelli di IOPS e throughput. PERSISTENT\_2 supporta l'archiviazione SSD e offre prestazioni superiori PerUnitStorageThroughput (fino a 1000 MB/s/TiB). PERSISTENT\_2 è disponibile in un numero limitato di Regioni AWS. Per ulteriori informazioni sui tipi di distribuzione e l'elenco delle aree Regioni AWS in cui PERSISTENT\_2 è disponibile, consulta [Opzioni di distribuzione del file system FSx per Lustre](#) nella Amazon FSx for Lustre User Guide.

La crittografia dei dati in transito viene abilitata automaticamente quando accedi SCRATCH\_2 o PERSISTENT\_2 distribuisce file system da istanze Amazon EC2 che supportano questa funzionalità. PERSISTENT\_1

La crittografia dei dati in transito e dei tipi di PERSISTENT\_2 distribuzione è supportata quando si accede da tipi di istanze supportati. SCRATCH\_2 PERSISTENT\_1 Regioni AWS Per ulteriori informazioni, [consulta Encrypting data in transit](#) nella Amazon FSx for Lustre User Guide.

 Note

Il supporto per il tipo di PERSISTENT\_2 distribuzione è stato aggiunto con la AWS ParallelCluster versione 3.2.0.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## ImportedFileChunkSize(Facoltativo,**Integer**)

Per i file importati da un archivio di dati, questo valore determina il numero di stripe e la quantità massima di dati per ogni file (in MiB) archiviato su un singolo disco fisico. Il numero massimo di dischi su cui un singolo file può essere distribuito è limitato al numero totale di dischi che compongono il file system.

La dimensione predefinita del blocco è 1.024 MiB (1 GiB), ma tale valore può raggiungere 512.000 MiB (500 GiB). Gli oggetti Amazon S3 hanno una dimensione massima di 5 TB.

### Note

Questo parametro non è supportato per i file system che utilizzano il `PERSISTENT_2` tipo di distribuzione. Per istruzioni su come configurare le associazioni degli archivi di dati, consulta [Collegamento del file system a un bucket S3 nella](#) Amazon FSx for Lustre User Guide.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## DataCompressionType(Facoltativo,) **String**

Imposta la configurazione di compressione dei dati FSx per il file system for Lustre. Il valore supportato è `LZ4`. `LZ4` indica che la compressione dei dati è attivata con l' `LZ4` algoritmo. Quando `DataCompressionType` non è specificato, la compressione dei dati viene disattivata al momento della creazione del file system.

Per ulteriori informazioni, vedete [Compressione dei dati Lustre](#).

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## ExportPath(Facoltativo,**String**)

Il percorso in Amazon S3 in cui viene esportata la radice del file system FSx for Lustre. Questa impostazione è supportata solo quando viene `ImportPath` specificato il parametro. Il percorso deve utilizzare lo stesso bucket Amazon S3 specificato in `ImportPath`. È possibile fornire un prefisso opzionale in cui esportare i dati nuovi e modificati dal file system FSx for Lustre. Se non viene fornito un `ExportPath` valore, FSx for Lustre imposta un percorso di esportazione predefinito, `s3://amzn-s3-demo-bucket/FSxLustre[creation-`

timestamp] Il timestamp è in formato UTC, ad esempio `s3://amzn-s3-demo-bucket/FSxLustre20181105T222312Z`.

Il bucket Amazon S3 di esportazione deve essere lo stesso bucket di importazione specificato da `ImportPath`. Se si specifica solo il nome di un bucket, ad esempio `s3://amzn-s3-demo-bucket`, si ottiene una mappatura 1:1 degli oggetti del file system agli oggetti bucket di Amazon S3. Questa mappatura significa che i dati di input in Amazon S3 vengono sovrascritti durante l'esportazione. Se fornisci un prefisso personalizzato nel percorso di esportazione, ad esempio `FSx per Lustre s3://amzn-s3-demo-bucket/[custom-optional-prefix]`, esporta il contenuto del tuo file system in quel prefisso di esportazione nel bucket Amazon S3.

#### Note

Questo parametro non è supportato per i file system che utilizzano il tipo di distribuzione. `PERSISTENT_2` Configura le associazioni degli archivi di dati come descritto in [Collegamento del file system a un bucket S3 nella Guida per l'utente di Amazon FSx for Lustre](#).

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### `ImportPath`(Facoltativo,) String

Il percorso del bucket Amazon S3 (incluso il prefisso opzionale) che stai utilizzando come repository di dati per il tuo FSx file system for Lustre. La radice del file system FSx for Lustre verrà mappata alla radice del bucket Amazon S3 selezionato. Un esempio è `s3://amzn-s3-demo-bucket/optional-prefix`. Se specifichi un nome dopo il nome del bucket Amazon S3, nel file system verranno caricate solo le chiavi degli oggetti con il prefisso specificato.

#### Note

Questo parametro non è supportato per i file system che utilizzano il tipo di distribuzione. `PERSISTENT_2` Configura le associazioni degli archivi di dati come descritto in [Collegamento del file system a un bucket S3 nella Guida per l'utente di Amazon FSx for Lustre](#).

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`WeeklyMaintenanceStartTime`(Facoltativo,) `String`

L'ora di inizio preferita per eseguire la manutenzione settimanale. È nel "d:HH:MM" formato del fuso orario UTC+0. In questo formato, d è il numero del giorno della settimana compreso tra 1 e 7, che inizia con il lunedì e termina con la domenica. Le virgolette sono obbligatorie per questo campo.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

`AutomaticBackupRetentionDays`(Facoltativo,**Integer**)

Numero di giorni per la retention dei backup automatici. Impostando questo valore su 0, i backup automatici vengono disattivati. L'intervallo supportato è compreso tra 0 e 90. Il valore predefinito è 0. Questa impostazione è valida solo per l'uso con i tipi `PERSISTENT_1` di `PERSISTENT_2` distribuzione. Per ulteriori informazioni, consulta [Lavorare con i backup nella Guida](#) per l'utente di Amazon FSx for Lustre.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

`CopyTagsToBackups`(Facoltativo,) `Boolean`

Se `true`, copia i tag del file system FSx for Lustre nei backup. Il valore predefinito è `false`. Se impostato su `true`, tutti i tag del file system, anche se non specificati dall'utente, vengono copiati in tutti i backup automatici e manuali. Se il valore è `true` e specifichi uno o più tag, solo i tag specificati vengono copiati nei backup. Se si specificano uno o più tag quando si crea un backup avviato dall'utente, nessun tag viene copiato dal file system, indipendentemente da questo valore. Questa impostazione è valida solo per l'uso con `PERSISTENT_1` e `PERSISTENT_2` i tipi di distribuzione.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`DailyAutomaticBackupStartTime`(Facoltativo,`String`)

Un'ora giornaliera ricorrente, nel HH:MM formato. HH è l'ora del giorno con numero zero (00-23). MM è il minuto dell'ora con numero zero (00-59). Ad esempio, `05:00` specifica le 5 del mattino ogni giorno. Questa impostazione è valida solo per l'uso con `PERSISTENT_1` e i tipi di `PERSISTENT_2` distribuzione.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

PerUnitStorageThroughput(Obbligatoria per **PERSISTENT\_1** i tipi di **PERSISTENT\_2** distribuzione,Integer)

Descrive la quantità di velocità effettiva di lettura e scrittura per ogni 1 tebibyte di storage, in MB/s/Tib. La capacità di throughput del file system viene calcolata moltiplicando la capacità di storage del file system (TiB) per il (of) PerUnitStorageThroughput restituisce il PerUnitStorageThroughput 120% del throughput MB/s/TiB). For a 2.4 TiB file system, provisioning 50 MB/s/TiB del file system. MB/s Si paga per la quantità di throughput fornita. Corrisponde alla proprietà. [PerUnitStorageThroughput](#)

Valori validi:

Archiviazione SSD PERSISTENT\_1:50, 100, 200 MB/s/Tib.

Archiviazione HDD PERSISTENT\_1:12, 40 MB/s/Tib.

Archiviazione SSD PERSISTENT\_2:125, 250, 500, 1000 MB/s/Tib.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

BackupId(StringOpzionale,)

Specifica l'ID del backup da utilizzare per ripristinare il file system FSx for Lustre da un backup esistente. Quando viene specificata l'BackupId impostazione, AutoImportPolicy, DeploymentType, ExportPath, KmsKeyId ImportPath ImportedFileChunkSizeStorageCapacity, e PerUnitStorageThroughput le impostazioni non devono essere specificate. Queste impostazioni vengono lette dal backup. Inoltre, le ImportedFileChunkSize impostazioni AutoImportPolicyExportPath, ImportPath, e non devono essere specificate. Corrisponde alla [BackupId](#) proprietà.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

KmsKeyId(Facoltativo,String)

L'ID della chiave AWS Key Management Service (AWS KMS) utilizzato per crittografare i dati del file system FSx for Lustre per renderli persistenti FSx per i file system Lustre inattivi. Se non specificato, viene utilizzata la chiave gestita FSx for Lustre. I file system SCRATCH\_1 and SCRATCH\_2 FSx for Lustre sono sempre crittografati quando sono inattivi utilizzando le chiavi gestite di FSx for Lustre. Per ulteriori informazioni, consulta [Encrypt nell'API Reference](#). AWS Key Management Service

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### FileSystemId(Facoltativo,**String**)

Specifica l'ID di un file system FSx for Lustre esistente.

Se viene specificata questa opzione, `FsxLustreSettings` vengono utilizzate solo le `FileSystemId` impostazioni `MountDir` e contenute in. Tutte le altre impostazioni di `FsxLustreSettings` vengono ignorate.

#### Note

Se viene utilizzato AWS Batch lo scheduler, FSx for Lustre è disponibile solo sul nodo principale.

#### Note

Il file system deve essere associato a un gruppo di sicurezza che consenta il traffico TCP in entrata e in uscita attraverso le porte 988, 1021, 1022 e 1023.

Assicurati che il traffico sia consentito tra il cluster e il file system effettuando una delle seguenti operazioni:

- Configura i gruppi di sicurezza del file system per consentire il traffico da e verso il CIDR o l'elenco di prefissi delle sottoreti del cluster.

#### Note

AWS ParallelCluster verifica che le porte siano aperte e che il CIDR o l'elenco dei prefissi sia configurato. AWS ParallelCluster non convalida il contenuto del blocco o dell'elenco di prefissi CIDR.

- Imposta gruppi di sicurezza personalizzati per i nodi del cluster utilizzando [SlurmQueues//SecurityGroupse NetworkingHeadNode/](#). [NetworkingSecurityGroups](#)  
I gruppi di sicurezza personalizzati devono essere configurati per consentire il traffico tra il cluster e il file system.

 Note

Se tutti i nodi del cluster utilizzano gruppi di sicurezza personalizzati, verifica AWS ParallelCluster solo che le porte siano aperte. AWS ParallelCluster non convalida che l'origine e la destinazione siano configurate correttamente.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### AutoImportPolicy(Facoltativo,**String**)

Quando crei il file system FSx for Lustre, gli oggetti Amazon S3 esistenti vengono visualizzati come elenchi di file e directory. Usa questa proprietà per scegliere in che modo FSx for Lustre mantiene aggiornati gli elenchi di file e directory man mano che aggiungi o modifichi oggetti nel bucket Amazon S3 collegato. AutoImportPolicy può avere i seguenti valori:

- **NEW**- L'importazione automatica è attiva. FSx for Lustre importa automaticamente gli elenchi di directory di tutti i nuovi oggetti aggiunti al bucket Amazon S3 collegato che attualmente non esistono nel FSx file system for Lustre.
- **NEW\_CHANGED**- L'importazione automatica è attiva. FSx for Lustre importa automaticamente gli elenchi di file e directory di tutti i nuovi oggetti aggiunti al bucket Amazon S3 e di tutti gli oggetti esistenti che vengono modificati nel bucket Amazon S3 dopo aver scelto questa opzione.
- **NEW\_CHANGED\_DELETED**- L'importazione automatica è attiva. FSx for Lustre importa automaticamente gli elenchi di file e directory di tutti i nuovi oggetti aggiunti al bucket Amazon S3, tutti gli oggetti esistenti che vengono modificati nel bucket Amazon S3 e tutti gli oggetti che sono stati eliminati nel bucket Amazon S3 dopo aver scelto questa opzione.

 Note

Il supporto per **NEW\_CHANGED\_DELETED** è stato aggiunto nella AWS ParallelCluster versione 3.1.1.

Se AutoImportPolicy non è specificato, l'importazione automatica è disattivata. FSx for Lustre aggiorna solo gli elenchi di file e directory dal bucket Amazon S3 collegato quando viene creato il file system. FSx for Lustre non aggiorna gli elenchi di file e directory per oggetti nuovi o modificati dopo aver scelto questa opzione.

Per ulteriori informazioni, consulta [Importa automaticamente gli aggiornamenti dal tuo bucket S3 nella Guida](#) per l'utente di Amazon FSx for Lustre.

 Note

Questo parametro non è supportato per i file system che utilizzano il PERSISTENT\_2 tipo di distribuzione. Per istruzioni su come configurare le associazioni degli archivi di dati, consulta [Collegamento del file system a un bucket S3 nella](#) Amazon FSx for Lustre User Guide.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

DriveCacheType(Facoltativo,) String

Specifica che il file system dispone di una cache dell'unità SSD. Questo può essere impostato solo se l'StorageType impostazione è impostata su e HDD l'DeploymentType impostazione è impostata su. PERSISTENT\_1 Corrisponde alla [DriveCacheType](#) proprietà. Per ulteriori informazioni, consulta FSx le [opzioni di distribuzione di Lustre](#) nella Amazon FSx for Lustre User Guide.

L'unico valore valido è READ. Per disabilitare la cache dell'unità SSD, non specificare l'impostazione. DriveCacheType

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

StorageType(Facoltativo,**String**)

Imposta il tipo di archiviazione FSx per il file system for Lustre che stai creando. I valori validi sono SSD e HDD.

- Imposta su SSD per utilizzare lo storage su unità a stato solido.
- Imposta per HDD utilizzare lo storage su disco rigido. HDD è supportato nei tipi di PERSISTENT distribuzione.

Il valore predefinito è SSD. Per ulteriori informazioni, consulta [Storage Type Options](#) nella Guida FSx per l'utente di Amazon per Windows e [Multiple Storage Options](#) nella Amazon FSx for Lustre User Guide.



Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### DeletionPolicy(Facoltativo,**String**)

Specifica se il file system deve essere mantenuto o eliminato quando il file system viene rimosso dal cluster o il cluster viene eliminato. I valori supportati sono `Delete` e `Retain`. Il valore predefinito è `Delete`.

Quando [DeletionPolicy](#) è impostato su `Delete`, un file system gestito, con i relativi dati, viene eliminato se il cluster viene eliminato o se il file system viene rimosso con un aggiornamento del cluster.

Per ulteriori informazioni, consulta [Archiviazione condivisa](#).

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

#### Note

`DeletionPolicy` è supportato a partire dalla AWS ParallelCluster versione 3.3.0.

### DataRepositoryAssociations(Facoltativo,**String**)

Elenco di DRAs (fino a 8 per file system)

A ogni associazione di repository di dati deve essere associata una directory FSx del file system Amazon univoca e un bucket o prefisso S3 univoco.

Non è possibile utilizzare [ExportPath](#) e [ImportPath](#) contemporaneamente. `FsxLustreSettings` DRAs

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### Name(Obbligatorio,**String**)

Il nome del DRA. Questo nome viene utilizzato quando si aggiornano le impostazioni.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### BatchImportMetaDataOnCreate(Facoltativo,**Boolean**)

Un flag booleano che indica se un'attività di importazione del repository di dati per importare i metadati deve essere eseguita dopo la creazione dell'associazione di repository di dati. L'attività viene eseguita se questo flag è impostato su `true`.

Valore predefinito: `false`

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`DataRepositoryPath`(Obbligatorio,String)

Il percorso del repository di dati Amazon S3 che verrà collegato al file system. Il percorso può essere un prefisso o un bucket S3 nel formato `s3://amzn-s3-demo-bucket/myPrefix/`. Questo percorso specifica la posizione del repository dati S3 in cui verranno importati o esportati i file.

Non può sovrapporsi ad altri DRAs

Modello: `^[^\u0000\u0085\u2028\u2029\r\n]{3,4357}$`

Minimo: 3

Massimo: 4357

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

`FileSystemPath`(Obbligatorio,String)

Un percorso sul file system Amazon FSx for Lustre che punta a una directory di alto livello (ad esempio/`ns1/`) o sottodirectory (come/`ns1/subdir/`) con cui verrà mappata 1-1. `DataRepositoryPath` La barra che precede il nome è obbligatoria. Due associazioni di repository di dati non possono avere percorsi di file system sovrapposti. Se ad esempio un repository di dati è associato al percorso del file system `/ns1/`, non è possibile collegare un altro repository di dati al percorso del file system `/ns1/ns2`.

Questo percorso specifica la posizione del file system in cui verranno esportati o importati i file. Questa directory del file system può essere collegata a un solo bucket Amazon S3 e nessun altro bucket S3 può essere collegato alla directory.

Non può sovrapporsi ad altri DRAs

 Note

Se indichi solo una barra (/) come percorso del file system, puoi collegare solo un repository di dati al file system. È possibile specificare "/" solo come percorso del file system per il primo repository di dati associato a un file system.

Modello: `^[^\u0000\u0085\u2028\u2029\r\n]{1,4096}$`

Minimo: 1

Massimo: 4096

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

`ImportedFileChunkSize`(Facoltativo,**Integer**)

Per i file importati da un repository di dati, questo valore stabilisce il conteggio delle stripe e la quantità massima di dati per file (in MiB) archiviati su un singolo disco fisico. Il numero massimo di dischi su cui un singolo file può essere distribuito è limitato al numero totale di dischi che compongono il file system o la cache.

La dimensione predefinita del blocco è 1.024 MiB (1 GiB), ma tale valore può raggiungere 512.000 MiB (500 GiB). Gli oggetti Amazon S3 hanno una dimensione massima di 5 TB.

Minimo: 1

Massimo: 4096

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

`AutoExportPolicy`(Facoltativo,Array of strings)

L'elenco può contenere uno o più dei seguenti valori:

- **NEW**: i nuovi file e le nuove directory vengono esportati automaticamente nel repository di dati man mano che vengono aggiunti al file system.
- **CHANGED**: le modifiche apportate ai file e alle directory del file system vengono esportate automaticamente nel repository di dati.
- **DELETED**: i file e le directory vengono eliminati automaticamente dal repository di dati quando vengono rimossi dal file system.

Puoi definire qualsiasi combinazione di tipi di eventi per `AutoExportPolicy`.

Massimo: 3

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

`AutoImportPolicy`(Facoltativo,Array of strings)

L'elenco può contenere uno o più dei seguenti valori:

- **NEW**- Amazon importa FSx automaticamente i metadati dei file aggiunti al bucket S3 collegato che attualmente non esistono nel FSx file system.
- **CHANGED**- Amazon aggiorna FSx automaticamente i metadati dei file e invalida il contenuto dei file esistenti sul file system quando i file vengono modificati nel repository di dati.
- **DELETED**- Amazon elimina FSx automaticamente i file dal file system man mano che i file corrispondenti vengono eliminati nell'archivio dati.

Puoi definire qualsiasi combinazione di tipi di eventi per `AutoImportPolicy`.

Massimo: 3

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

## FsxOntapSettings

### Note

È necessario definire `FsxOntapSettings` se `FsxOntap` è specificato per. [StorageType](#)

(Facoltativo) Le impostazioni per un file system FSx for ONTAP.

[FsxOntapSettings](#):

[VolumeId](#): *string*

## Proprietà `FsxOntapSettings`

`VolumeId`(Obbligatorio,String)

Specifica l'ID del volume del sistema esistente FSx for ONTAP.

 Note

- Se viene utilizzato uno AWS Batch scheduler, FSx for ONTAP è disponibile solo sul nodo principale.
- Se il tipo di distribuzione FSx for ONTAP è `Multi-AZ`, assicurati che la tabella di routing della sottorete del nodo principale sia configurata correttamente.
- Il supporto FSx per ONTAP è stato aggiunto nella AWS ParallelCluster versione 3.2.0.
- Il file system deve essere associato a un gruppo di sicurezza che consenta il traffico TCP e UDP in entrata e in uscita attraverso le porte 111, 635, 2049 e 4046.

Assicurati che il traffico sia consentito tra il cluster e il file system eseguendo una delle seguenti azioni:

- Configura i gruppi di sicurezza del file system per consentire il traffico da e verso il CIDR o l'elenco di prefissi delle sottoreti del cluster.

 Note

AWS ParallelCluster verifica che le porte siano aperte e che il CIDR o l'elenco dei prefissi sia configurato. AWS ParallelCluster non convalida il contenuto del blocco o dell'elenco di prefissi CIDR.

- Imposta gruppi di sicurezza personalizzati per i nodi del cluster utilizzando [SlurmQueues//SecurityGroupse NetworkingHeadNode/](#). [NetworkingSecurityGroups](#) I gruppi di sicurezza personalizzati devono essere configurati per consentire il traffico tra il cluster e il file system.

 Note

Se tutti i nodi del cluster utilizzano gruppi di sicurezza personalizzati, verifica AWS ParallelCluster solo che le porte siano aperte. AWS ParallelCluster non convalida che l'origine e la destinazione siano configurate correttamente.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## FsxOpenZfsSettings

### Note

È necessario definire `FsxOpenZfsSettings` se `FsxOpenZfs` è specificato per [StorageType](#).

(Facoltativo) Le impostazioni per un file system FSx per OpenZFS.

[FsxOpenZfsSettings](#):

[VolumeId](#): *string*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Proprietà **FsxOpenZfsSettings**

`VolumeId`(Obbligatorio,) `String`

Specifica l'ID del volume del sistema esistente FSx per OpenZFS.

### Note

- Se viene utilizzato uno AWS Batch scheduler, FSx per OpenZFS è disponibile solo sul nodo principale.
- Il supporto FSx per OpenZFS è stato aggiunto nella AWS ParallelCluster versione 3.2.0.
- Il file system deve essere associato a un gruppo di sicurezza che consenta il traffico TCP e UDP in entrata e in uscita attraverso le porte 111, 2049, 20001, 20002 e 20003.

Assicurati che il traffico sia consentito tra il cluster e il file system effettuando una delle seguenti operazioni:

- Configura i gruppi di sicurezza del file system per consentire il traffico da e verso il CIDR o l'elenco di prefissi delle sottoreti del cluster.

**Note**

AWS ParallelCluster verifica che le porte siano aperte e che il CIDR o l'elenco dei prefissi sia configurato. AWS ParallelCluster non convalida il contenuto del blocco o dell'elenco di prefissi CIDR.

- Imposta gruppi di sicurezza personalizzati per i nodi del cluster utilizzando [SlurmQueues//SecurityGroupse NetworkingHeadNode/](#). [NetworkingSecurityGroups](#) I gruppi di sicurezza personalizzati devono essere configurati per consentire il traffico tra il cluster e il file system.

**Note**

Se tutti i nodi del cluster utilizzano gruppi di sicurezza personalizzati, verifica AWS ParallelCluster solo che le porte siano aperte. AWS ParallelCluster non convalida che l'origine e la destinazione siano configurate correttamente.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## FileCacheSettings

**Note**

È necessario definire FileCacheSettings se FileCache è specificato per [StorageType](#).

(Facoltativo) Le impostazioni per una cache di file.

[FileCacheSettings](#):

[FileCacheId](#): *string*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Proprietà **FileCacheSettings**

**FileCacheId**(Obbligatorio,String)

Specifica l'ID della cache dei file di una cache di file esistente.

### Note

- File Cache non supporta gli AWS Batch scheduler.
- Il supporto per File Cache è stato aggiunto nella AWS ParallelCluster versione 3.7.0.
- Il file system deve essere associato a un gruppo di sicurezza che consenta il traffico TCP in entrata e in uscita attraverso la porta 988.

Assicurati che il traffico sia consentito tra il cluster e il file system effettuando una delle seguenti operazioni:

- Configura i gruppi di sicurezza della File Cache per consentire il traffico da e verso il CIDR o l'elenco di prefissi delle sottoreti del cluster.

### Note

AWS ParallelCluster verifica che le porte siano aperte e che il CIDR o l'elenco dei prefissi sia configurato. AWS ParallelCluster non convalida il contenuto del blocco o dell'elenco di prefissi CIDR.

- Imposta gruppi di sicurezza personalizzati per i nodi del cluster utilizzando [SlurmQueues//SecurityGroupse NetworkingHeadNode/](#). [NetworkingSecurityGroups](#) I gruppi di sicurezza personalizzati devono essere configurati per consentire il traffico tra il cluster e il file system.

### Note

Se tutti i nodi del cluster utilizzano gruppi di sicurezza personalizzati, verifica AWS ParallelCluster solo che le porte siano aperte. AWS ParallelCluster non convalida che l'origine e la destinazione siano configurate correttamente.



Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Sezione **Iam**

(Facoltativo) Specifica le proprietà IAM per il cluster.

```
Iam:
 Roles:
 LambdaFunctionsRole: string
 PermissionsBoundary: string
 ResourcePrefix: string
```

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

### Proprietà **Iam**

PermissionsBoundary(Facoltativo,**String**)

L'ARN della policy IAM da utilizzare come limite delle autorizzazioni per tutti i ruoli creati da AWS ParallelCluster. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM. Il formato è `arn:${Partition}:iam:${Account}:policy/${PolicyName}`.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Roles(Facoltativo)

Specifica le impostazioni per i ruoli IAM utilizzati dal cluster.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

LambdaFunctionsRole(Facoltativo,**String**)

L'ARN del ruolo IAM per cui utilizzare. AWS Lambda Questo sostituisce il ruolo predefinito associato a tutte le funzioni Lambda che supportano le risorse personalizzate. AWS CloudFormation Lambda deve essere configurata come principale autorizzata ad assumere il ruolo. Ciò non sostituirà il ruolo delle funzioni Lambda utilizzate per AWS Batch. Il formato è `arn:${Partition}:iam:${Account}:role/${RoleName}`.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## ResourcePrefix(Facoltativo)

Specifica un percorso o un prefisso del nome per le risorse IAM create da AWS ParallelCluster.

Il prefisso della risorsa deve seguire le [regole di denominazione](#) specificate da IAM:

- Un nome può contenere fino a 30 caratteri.
- Un nome può essere solo una stringa senza caratteri slash (/).
- Un percorso può contenere fino a 512 caratteri.
- Un percorso deve iniziare e terminare con una barra (/). Può contenere più barre (/) tra la barra iniziale e quella finale (/).
- È possibile combinare il percorso e il nome. /path/name

Specificate un nome.

```
Iam:
 ResourcePrefix: my-prefix
```

Specificate un percorso.

```
Iam:
 ResourcePrefix: /org/dept/team/project/user/
```

Specificate un percorso e un nome.

```
Iam:
 ResourcePrefix: /org/dept/team/project/user/my-prefix
```

Se si specifica /my-prefix, viene restituito un errore.

```
Iam:
 ResourcePrefix: /my-prefix
```

Viene restituito un errore di configurazione. Un percorso deve avere due / s. Un prefisso di per sé non può avere / s.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## Sezione **LoginNodes**

### Note

Il supporto per LoginNodes è stato aggiunto nella AWS ParallelCluster versione 3.7.0.

(Facoltativo) Specifica la configurazione per il pool di nodi di accesso.

```
LoginNodes:
 Pools:
 - Name: string
 Count: integer
 InstanceType: string
 GracetimePeriod: integer
 Image:
 CustomAmi: string
 Ssh:
 KeyName: string
 AllowedIps: string
 Networking:
 SubnetIds:
 - string
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 Dcv:
 Enabled: boolean
 Port: integer
 AllowedIps: string
 CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
```

```

 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string

```

Politica di aggiornamento: i nodi di accesso nel cluster devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

Proprietà **LoginNodes**

Proprietà **Pools**

Definisce gruppi di nodi di accesso con la stessa configurazione di risorse. A partire dalla AWS ParallelCluster versione 3.11.0 è possibile specificare fino a 10 pool.

```

Pools:
 - Name: string
 Count: integer
 InstanceType: string
 GracetimePeriod: integer
 Image:
 CustomAmi: string
 Ssh:
 KeyName: string
 AllowedIps: string
 Networking:
 SubnetIds:
 - string
 SecurityGroups:

```

```
- string
AdditionalSecurityGroups:
- string
Dcv:
 Enabled: boolean
 Port: integer
 AllowedIps: string
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string
```

Politica di aggiornamento: è possibile aggiungere pool di nodi di accesso, ma la rimozione di un pool richiede l'interruzione di tutti i nodi di accesso nel cluster.

## Name(Obbligatorio) String

Specificate il nome del LoginNodes pool. Viene utilizzato per etichettare le LoginNodes risorse.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Note

A partire dalla AWS ParallelCluster versione 3.11.0, la politica di aggiornamento è: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione venga modificata per un aggiornamento.

## Count(Obbligatorio) Integer

Specifica il numero di nodi di accesso da mantenere attivi.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## InstanceType(Obbligatorio) String

Specifica il tipo di istanza Amazon EC2 utilizzato per il nodo di accesso. L'architettura del tipo di istanza deve essere la stessa dell'architettura utilizzata per Slurm InstanceType l'impostazione.

Politica di aggiornamento: questa impostazione può essere modificata se il pool di nodi di accesso viene interrotto.

### Note

A partire dalla AWS ParallelCluster versione 3.11.0, la politica di aggiornamento è: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

## GracetimePeriod(Facoltativo) Integer

Specifica il periodo di tempo minimo, in minuti, che intercorre tra la notifica all'utente connesso che un nodo di accesso deve essere disattivato e l'evento di arresto effettivo. I valori validi per GracetimePeriod sono compresi tra 3 e 120 minuti. L'impostazione predefinita è 10 minuti.

 Note

L'evento di attivazione implica interazioni tra più AWS servizi. A volte, la latenza di rete e la propagazione delle informazioni potrebbero richiedere del tempo, pertanto il periodo di prova potrebbe richiedere più tempo del previsto a causa di ritardi interni nei servizi. AWS

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Image(Facoltativo)

Definisce la configurazione dell'immagine per i nodi di accesso.

Image:

CustomAmi: *String*

## CustomAmi(FacoltativoString)

Specifica l'AMI personalizzata utilizzata per effettuare il provisioning dei nodi di accesso. Se non viene specificato, il valore predefinito è quello specificato in. [Sezione HeadNode](#)

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Ssh(Facoltativo)

Definisce la ssh configurazione per i nodi di accesso.

Ssh:

KeyName: *string*

AllowedIps: *string*

 Note

A partire dalla AWS ParallelCluster versione 3.11.0, la politica di aggiornamento è: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione venga modificata per un aggiornamento.

## KeyName(Facoltativo) String

Specifica la ssh chiave utilizzata per accedere ai nodi di accesso. Se non viene specificato, il valore predefinito è quello specificato in. [Sezione HeadNode](#)

Politica di aggiornamento: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

### Important

Obsoleto: il parametro di configurazione LoginNodes/PoolSsh/KeyName è obsoleto e verrà rimosso nelle versioni future. La CLI ora restituisce un messaggio di avviso quando viene utilizzata nella configurazione del cluster. Vedi <https://github.com/aws/aws-parallelcluster/issues/6811> per i dettagli.

## AllowedIps(Facoltativo) String

Specifica l'intervallo IP in formato CIDR o un ID dell'elenco di prefissi per le connessioni SSH ai nodi di accesso nel pool. L'impostazione predefinita è quella [AllowedIps](#) definita nella configurazione del nodo principale o se non specificata. 0.0.0.0/0 [Sezione HeadNode](#).

Politica di aggiornamento: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

### Note

Il supporto AllowedIps per i nodi di accesso è stato aggiunto nella AWS ParallelCluster versione 3.11.0.

## Networking(Obbligatorio)

### Networking:

#### SubnetIds:

- *string*

#### SecurityGroups:

- *string*

#### AdditionalSecurityGroups:



- *string*

#### Note

A partire dalla AWS ParallelCluster versione 3.11.0, la politica di aggiornamento è: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione possa essere modificata per un aggiornamento.

### SubnetIds(Obbligatorio) [String]

L'ID della sottorete esistente in cui si effettua il provisioning del pool di nodi di accesso. È possibile definire solo una sottorete.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### SecurityGroups(Facoltativo) [String]

Un elenco di gruppi di sicurezza da utilizzare per il pool di nodi di accesso. Se non viene specificato alcun gruppo di sicurezza, AWS ParallelCluster crea gruppi di sicurezza per te.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### AdditionalSecurityGroups(Facoltativo[String])

Un elenco di gruppi di sicurezza aggiuntivi da utilizzare per il pool di nodi di accesso.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Dcv(Facoltativo)

Definisce le impostazioni di configurazione per il server NICE DCV che viene eseguito sui nodi di [accesso](#). Per ulteriori informazioni, consulta [Connect ai nodi principali e di accesso tramite Amazon DCV](#)

#### Dcv:

Enabled: *boolean*

Port: *integer*

AllowedIps: *string*

 Important

Per impostazione predefinita, la porta NICE DCV configurata da AWS ParallelCluster è aperta a tutti gli IPv4 indirizzi. Puoi connetterti a una porta NICE DCV solo se hai l'URL per la sessione NICE DCV e connetterti alla sessione NICE DCV entro 30 secondi da quando l'URL viene restituito da `pcluster dcv-connect`. Utilizzate l'`AllowedIps` impostazione per limitare ulteriormente l'accesso alla porta NICE DCV con un intervallo IP in formato CIDR e utilizzate l'impostazione `Porta` per impostare una porta non standard.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

 Note

Il supporto per DCV sui nodi di accesso è stato aggiunto nella AWS ParallelCluster versione 3.11.0.

### Enabled(Richiesto) Boolean

Specifica se NICE DCV è abilitato sui nodi di accesso del pool. Il valore predefinito è `false`.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

 Note

NICE DCV genera automaticamente un certificato autofirmato che viene utilizzato per proteggere il traffico tra il client NICE DCV e il server NICE DCV che viene eseguito sul nodo di accesso. Per configurare il proprio certificato, consulta [Certificato HTTPS Amazon DCV](#).

### Port(Facoltativo) Integer

Specifica la porta per NICE DCV. Il valore predefinito è 8443.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### AllowedIps(Facoltativo) String

Specifica l'intervallo IP in formato CIDR per le connessioni a NICE DCV. Questa impostazione viene utilizzata solo quando si crea il gruppo di sicurezza. AWS ParallelCluster Il valore predefinito è `0.0.0.0/0`, che consente l'accesso da qualsiasi indirizzo Internet.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### CustomActions(Facoltativo)

Speciifica gli script personalizzati da eseguire sui nodi di accesso.

```
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
```

 Note

Il supporto per le azioni personalizzate sui nodi di accesso è stato aggiunto nella AWS ParallelCluster versione 3.11.0.

### OnNodeStart(Facoltativo)

Specifica un singolo script o una sequenza di script da eseguire sui [nodi di accesso](#) prima che venga avviata qualsiasi azione di bootstrap di distribuzione del nodo. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

#### Sequence(Facoltativo)

Elenco degli script da eseguire. AWS ParallelCluster esegue gli script nello stesso ordine in cui sono elencati nel file di configurazione, a partire dal primo.

#### Script(ObbligatorioString)

Specificate il file da utilizzare. Il percorso del file può iniziare con `https:// os3://`.

#### Args(Facoltativo[String])

Elenco di argomenti da passare allo script.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

#### Script(ObbligatorioString)

Specifica il file da utilizzare per un singolo script. Il percorso del file può iniziare con `https:// os3://`.

#### Args(Facoltativo[String])

Elenco di argomenti da passare al singolo script.

### OnNodeConfigured(Facoltativo)

Specifica un singolo script o una sequenza di script da eseguire sui [nodi di accesso dopo il completamento dei](#) processi di bootstrap del nodo. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

## Sequence(Facoltativo)

Elenco degli script da eseguire. AWS ParallelCluster esegue gli script nello stesso ordine in cui sono elencati nel file di configurazione, a partire dal primo.

Script(ObbligatorioString)

Specificate il file da utilizzare. Il percorso del file può iniziare con `https://os3://`.

Args(Facoltativo[String])

Elenco di argomenti da passare allo script.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

Script(ObbligatorioString)

Specifica il file da utilizzare per un singolo script. Il percorso del file può iniziare con `https://os3://`.

Args(Facoltativo[String])

Elenco di argomenti da passare al singolo script.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

OnNodeUpdated(Facoltativo)

Specifica un singolo script o una sequenza di script da eseguire dopo il completamento dell'aggiornamento del nodo principale e l'allineamento dello scheduler e dello storage condiviso con le ultime modifiche alla configurazione del cluster. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

Sequence(Facoltativo)

Elenco degli script da eseguire. AWS ParallelCluster esegue gli script nello stesso ordine in cui sono elencati nel file di configurazione, a partire dal primo.

Script(ObbligatorioString)

Specificate il file da utilizzare. Il percorso del file può iniziare con `https://os3://`.

Args(Facoltativo[String])

Elenco di argomenti da passare allo script.

## Script(ObbligatorioString)

Specifica il file da utilizzare per un singolo script. Il percorso del file può iniziare con `https://` o `s3://`.

## Args(Facoltativo[String])

Elenco di argomenti da passare al singolo script.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Note

AWS ParallelCluster non supporta l'inclusione sia di un singolo script che della stessa azione personalizzata. Sequence

## Iam(Facoltativo)

Specifica un ruolo di istanza o un profilo di istanza da utilizzare sui nodi di accesso per sovrascrivere il ruolo o il profilo di istanza di istanza predefinito per il cluster.

### Iam:

```
InstanceRole: string
InstanceProfile: string
AdditionalIamPolicies:
- Policy: string
```

### Note

A partire dalla AWS ParallelCluster versione 3.11.0, la politica di aggiornamento è: i nodi di accesso nel pool devono essere interrotti affinché questa impostazione venga modificata per un aggiornamento.

## InstanceProfile(Facoltativo) String

Specificate un profilo di istanza per sovrascrivere il profilo di istanza del nodo di accesso predefinito. Non è possibile specificare InstanceProfile e InstanceRole. Il formato è `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

Se viene specificato, le `AdditionalIamPolicies` impostazioni `InstanceRole` and non possono essere specificate.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### `InstanceRole(FacoltativoString)`

Speciifica un ruolo di istanza per sovrascrivere il ruolo di istanza del nodo di accesso predefinito. Non è possibile specificare `InstanceProfile` e `InstanceRole`. Il formato è `arn:Partition:iam::Account:role/RoleName`. Se viene specificato, le `AdditionalIamPolicies` impostazioni `InstanceProfile` and non possono essere specificate.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### `AdditionalIamPolicies(Facoltativo)`

`AdditionalIamPolicies`:

- `Policy`: *string*

Una politica IAM Amazon Resource Name (ARN).

Speciifica un elenco di Amazon Resource Names (ARN) delle policy IAM per Amazon EC2. Questo elenco è allegato al ruolo root utilizzato per il nodo di accesso oltre alle autorizzazioni richieste da AWS ParallelCluster

Il nome di una policy IAM e il relativo ARN sono diversi. I nomi non possono essere usati.

Se è specificato, le `InstanceRole` impostazioni `InstanceProfile` and non possono essere specificate. Si consiglia di utilizzarle `AdditionalIamPolicies` perché `AdditionalIamPolicies` vengono aggiunte alle autorizzazioni richieste e `InstanceRole` devono includere tutte le autorizzazioni richieste. AWS ParallelCluster Le autorizzazioni richieste cambiano spesso da una versione all'altra man mano che vengono aggiunte funzionalità.

Non esiste un valore predefinito.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

Policy(Obbligatorio[String])

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

## Sezione **Monitoring**

(Facoltativo) Specificate le impostazioni di monitoraggio per il cluster.

Monitoring:

Logs:

CloudWatch:

Enabled: *boolean*

RetentionInDays: *integer*

DeletionPolicy: *string*

Rotation:

Enabled: *boolean*

Dashboards:

CloudWatch:

Enabled: *boolean*

DetailedMonitoring: *boolean*

Alarms:

Enabled: *boolean*

Politica di aggiornamento: questa impostazione non viene analizzata durante un aggiornamento.

## Proprietà **Monitoring**

Logs(Facoltativo)

Le impostazioni del registro per il cluster.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

CloudWatch(Facoltativo)

Le impostazioni CloudWatch dei log per il cluster.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.



## Enabled(Obbligatorio,Boolean)

Set true, i log del cluster vengono trasmessi in streaming a CloudWatch Logs. Il valore predefinito è true.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## RetentionInDays(Facoltativo,) Integer

Il numero di giorni in cui conservare gli eventi di registro nei CloudWatch registri. Il valore predefinito è 180. I valori supportati sono 0, 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827 e 3653. Un valore pari a 0 utilizzerà l'impostazione di conservazione dei CloudWatch log predefinita, ovvero non scadrà mai.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

## DeletionPolicy(Facoltativo,String)

Indica se eliminare gli eventi di registro CloudWatch nei registri quando il cluster viene eliminato. I valori possibili sono Delete e Retain. Il valore predefinito è Retain.

[Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.](#)

## Rotation(Facoltativo)

Le impostazioni di rotazione dei log per il cluster.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## Enabled(Obbligatorio,Boolean)

Set true, la rotazione dei log è abilitata. Il valore predefinito è true. Quando un file di registro AWS ParallelCluster configurato raggiunge una certa dimensione, viene ruotato e viene mantenuto un unico backup. Per ulteriori informazioni, consulta [AWS ParallelCluster rotazione dei log configurata](#).

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

## Dashboards(Facoltativo)

Le impostazioni del dashboard per il cluster.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## CloudWatch(Facoltativo)

Le impostazioni del CloudWatch dashboard per il cluster.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Enabled(Obbligatorio,Boolean)

Settrue, la CloudWatch dashboard è abilitata. Il valore predefinito è true.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## DetailedMonitoring(Facoltativo,Boolean)

Se impostato su true, il monitoraggio dettagliato è abilitato per la flotta di calcolo delle istanze Amazon EC2. Se abilitata, la console Amazon EC2 visualizza grafici per il monitoraggio delle istanze a intervalli di 1 minuto. Ci sono costi aggiuntivi quando questa funzionalità è abilitata. Il valore predefinito è false.

Per ulteriori informazioni, consulta [Abilitazione o disabilitazione del monitoraggio dettagliato per le istanze](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

### Note

DetailedMonitoringviene aggiunto a partire dalla AWS ParallelCluster versione 3.6.0.

## Alarms(Facoltativo)

CloudWatch Allarmi per il cluster.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Enabled(Opzionale)

Set true, verranno creati gli CloudWatch allarmi per il cluster. Il valore predefinito è true.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

#### Note

A partire dalla AWS ParallelCluster versione 3.8.0, vengono creati i seguenti allarmi per il nodo principale: Amazon EC2 Health Check, CPU/Memory/Disk utilizzo e un allarme composito che include tutti gli altri.

## Sezione **Tags**

(Facoltativo), Array Definisce i tag utilizzati CloudFormation e propagati a tutte le risorse del cluster. Per ulteriori informazioni, consulta il [tag CloudFormation delle risorse](#) nella Guida per l'AWS CloudFormation utente.

#### Tags:

- Key: *string*
- Value: *string*

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

## Proprietà **Tags**

Key(Obbligatorio,String)

Definisce il nome del tag.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

Value(Obbligatorio,String)

Definisce il valore del tag.

Politica di aggiornamento: questa impostazione può essere modificata durante un aggiornamento.

**Note**

A partire dalla AWS ParallelCluster versione 3.15.0, gli aggiornamenti dei tag sono supportati con le seguenti limitazioni:

- Volume EBS attivo HeadNode : conserverà solo i tag di quando è stato creato il cluster; l'aggiornamento dei tag su questo volume EBS non è supportato.
- Running Nodes: gli aggiornamenti dei tag non verranno applicati ai nodi di calcolo o di accesso in esecuzione.

## Sezione **AdditionalPackages**

(Facoltativo) Utilizzato per identificare i pacchetti aggiuntivi da installare.

AdditionalPackages:

IntelSoftware:

IntelHpcPlatform: *boolean*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### **IntelSoftware**

(Facoltativo) Definisce la configurazione per le soluzioni Intel Select.

IntelSoftware:

IntelHpcPlatform: *boolean*

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Proprietà **IntelSoftware**

IntelHpcPlatform(Facoltativo,Boolean)

Settrue, indica che il [contratto di licenza con l'utente finale](#) per Intel Parallel Studio è accettato. Ciò fa sì che Intel Parallel Studio venga installato sul nodo principale e condiviso con i nodi di elaborazione. Ciò aggiunge diversi minuti al tempo impiegato dal nodo principale per avviarsi.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

 Note

A partire dalla AWS ParallelCluster versione 3.10.0 il `IntelHpcPlatform` parametro non è più supportato.

## Sezione **DirectoryService**

 Note

Il supporto per `DirectoryService` è stato aggiunto nella AWS ParallelCluster versione 3.1.1.

(Facoltativo) Le impostazioni del servizio di directory per un cluster che supporta l'accesso di più utenti.

AWS ParallelCluster gestisce le autorizzazioni che supportano l'accesso di più utenti ai cluster con Active Directory (AD) tramite Lightweight Directory Access Protocol (LDAP) supportato dal [System Security Services Daemon](#) (SSSD). [Per ulteriori informazioni, consulta Cos'è? AWS Directory Service](#) nella Guida all'AWS Directory Service amministrazione.

Si consiglia di utilizzare LDAP over TLS/SSL (abbreviato in LDAPS) per garantire che tutte le informazioni potenzialmente sensibili vengano trasmesse su canali crittografati.

`DirectoryService`:

```
DomainName: string
DomainAddr: string
PasswordSecretArn: string
DomainReadOnlyUser: string
LdapTlsCaCert: string
LdapTlsReqCert: string
LdapAccessFilter: string
GenerateSshKeysForUsers: boolean
AdditionalSssdConfigs: dict
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

## Proprietà **DirectoryService**

### Note

Se prevedi di utilizzarlo AWS ParallelCluster in un'unica sottorete senza accesso a Internet, consulta [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#) la sezione dedicata ai requisiti aggiuntivi.

### DomainName(Obbligatorio,String)

Il dominio Active Directory (AD) utilizzato per le informazioni sull'identità.

DomainName accetta i formati Fully Qualified Domain Name (FQDN) e LDAP Distinguished Name (DN).

- Esempio di FQDN: corp.*example*.com
- Esempio di DN LDAP: DC=*corp*,DC=*example*,DC=*com*

Questa proprietà corrisponde al parametro sssd-ldap chiamato. ldap\_search\_base

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

### DomainAddr(Obbligatorio,) String

L'URI o URIs quel punto rimanda al controller di dominio AD utilizzato come server LDAP. L'URI corrisponde al parametro SSSD-LDAP chiamato. ldap\_uri Il valore può essere una stringa separata da virgole di. URIs Per utilizzare LDAP, è necessario ldap:// aggiungere all'inizio di ogni URI.

Valori di esempio:

```
ldap://192.0.2.0,ldap://203.0.113.0 # LDAP
ldaps://192.0.2.0,ldaps://203.0.113.0 # LDAPS without support for certificate
verification
ldaps://abcdef01234567890.corp.example.com # LDAPS with support for certificate
verification
```

```
192.0.2.0,203.0.113.0
default
```

```
AWS ParallelCluster uses LDAPS by
```

Se si utilizza LDAPS con la verifica del certificato, URIs devono essere nomi host.

Se si utilizza LDAPS senza verifica del certificato o LDAP, URIs possono essere nomi host o indirizzi IP.

Utilizzate LDAP over TLS/SSL (LDAPS) per evitare la trasmissione di password e altre informazioni sensibili su canali non crittografati. Se AWS ParallelCluster non trova un protocollo, lo aggiunge `ldaps://` all'inizio di ogni URI o nome host.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

`PasswordSecretArn`(Obbligatorio,String)

L'Amazon Resource Name (ARN) del Gestione dei segreti AWS segreto che contiene la password in `clearDomainReadOnlyUser`. Il contenuto del segreto corrisponde al parametro SSSD-LDAP chiamato `ldap_default_authtok`

 Note

Quando usi la Gestione dei segreti AWS console per creare un segreto, assicurati di selezionare «Altro tipo di segreto», seleziona il testo non crittografato e includi solo il testo della password nel segreto.

Per ulteriori informazioni su come Gestione dei segreti AWS creare un segreto, consulta [Create](#) an Secret Gestione dei segreti AWS

Il client LDAP utilizza la password per l'autenticazione nel dominio AD `DomainReadOnlyUser` quando richiede informazioni sull'identità.

Se l'utente ha il permesso di farlo [DescribeSecret](#), `PasswordSecretArn` viene convalidato. `PasswordSecretArn` è valido se il segreto specificato esiste. Se la policy IAM dell'utente non include `DescribeSecret`, `PasswordSecretArn` non viene convalidata e viene visualizzato un messaggio di avviso. Per ulteriori informazioni, consulta [Politica AWS ParallelCluster pcluster utente di base](#).

Quando il valore del segreto cambia, il cluster non viene aggiornato automaticamente. Per aggiornare il cluster in base al nuovo valore segreto, è necessario interrompere la flotta di calcolo

con il [the section called “pcluster update-compute-fleet”](#) comando e quindi eseguire il comando seguente dall'interno del nodo principale.

```
$ sudo /opt/parallelcluster/scripts/directory_service/
update_directory_service_password.sh
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

DomainReadOnlyUser(Obligatorio,String)

L'identità utilizzata per interrogare il dominio AD per ottenere informazioni sull'identità durante l'autenticazione degli accessi degli utenti del cluster. Corrisponde al parametro SSSD-LDAP chiamato. `ldap_default_bind_dn` Usa le informazioni sulla tua identità AD per questo valore.

Specificate l'identità nel modulo richiesto dallo specifico client LDAP presente sul nodo:

- Microsoft AD:

```
cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

- Annuncio semplice:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

LdapTlsCaCert(Facoltativo,) String

Il percorso assoluto di un pacchetto di certificati che contiene i certificati di ogni autorità di certificazione nella catena di certificazione che ha emesso un certificato per i controller di dominio. Corrisponde al parametro SSSD-LDAP chiamato. `ldap_tls_cacert`

Un pacchetto di certificati è un file composto dalla concatenazione di certificati distinti in formato PEM, noto anche come formato DER Base64 in Windows. Viene utilizzato per verificare l'identità del controller di dominio AD che funge da server LDAP.

AWS ParallelCluster non è responsabile del posizionamento iniziale dei certificati sui nodi. In qualità di amministratore del cluster, puoi configurare manualmente il certificato nel nodo principale dopo la creazione del cluster oppure puoi utilizzare uno [script di bootstrap](#). In



alternativa, puoi utilizzare un'Amazon Machine Image (AMI) che include il certificato configurato sul nodo principale.

[Simple AD](#) non fornisce il supporto LDAPS. Per informazioni su come integrare una directory Simple AD con AWS ParallelCluster, vedi [Come configurare un endpoint LDAPS per Simple AD](#) nel blog sulla AWS sicurezza.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

`LdapTlsReqCert(Facoltativo,)` `String`

Specifica quali controlli eseguire sui certificati del server in una sessione TLS. Corrisponde al parametro SSSD-LDAP chiamato `ldap_tls_reqcert`

Valori validi: `never`, `allow`, `try`, `demand` e `hard`.

`neverallow`, e `try` abilita il proseguimento delle connessioni anche se vengono rilevati problemi con i certificati.

`demand` e `hard` abilita la continuazione della comunicazione se non vengono rilevati problemi con i certificati.

Se l'amministratore del cluster utilizza un valore che non richiede la convalida del certificato per avere esito positivo, viene restituito un messaggio di avviso all'amministratore. Per motivi di sicurezza, ti consigliamo di non disabilitare la verifica dei certificati.

Il valore predefinito è `hard`.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

`LdapAccessFilter(Facoltativo,String)`

Specificate un filtro per limitare l'accesso alla directory a un sottoinsieme di utenti. Questa proprietà corrisponde al parametro SSSD-LDAP chiamato `ldap_access_filter`. È possibile utilizzarlo per limitare le query a un AD che supporta un numero elevato di utenti.

Questo filtro può bloccare l'accesso degli utenti al cluster. Tuttavia, non influisce sulla reperibilità degli utenti bloccati.

Se questa proprietà è impostata, il parametro SSSD `access_provider` è impostato `ldap` internamente da AWS ParallelCluster e non deve essere modificato da [DirectoryService/](#) [settings](#). [AdditionalSssdConfigs](#)

Se questa proprietà viene omessa e l'accesso utente personalizzato non è specificato in [DirectoryService/AdditionalSsdConfigs](#), tutti gli utenti della directory possono accedere al cluster.

Esempi:

```
!"(cn=SomeUser*)" # denies access to every user with an alias that starts with
"SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias that starts with
"SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".
```

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

**GenerateSshKeysForUsers**(Facoltativo,**Boolean**)

Definisce se AWS ParallelCluster genera una chiave SSH per gli utenti del cluster immediatamente dopo l'autenticazione iniziale sul nodo principale.

Se impostato su `true`, una chiave SSH viene generata e salvata `USER_HOME_DIRECTORY/.ssh/id_rsa`, se non esiste, per ogni utente dopo la prima autenticazione sul nodo principale.

Per un utente che non è ancora stato autenticato sul nodo principale, la prima autenticazione può avvenire nei seguenti casi:

- L'utente accede al nodo principale per la prima volta con la propria password.
- Nel nodo principale, un sudoer passa all'utente per la prima volta: su `USERNAME`
- Nel nodo principale, un sudoer esegue un comando come utente per la prima volta: su `-u USERNAME COMMAND`

Gli utenti possono utilizzare la chiave SSH per gli accessi successivi al nodo principale del cluster e ai nodi di calcolo. Con AWS ParallelCluster, gli accessi tramite password ai nodi di calcolo del cluster sono disabilitati in base alla progettazione. Se un utente non ha effettuato l'accesso al nodo principale, le chiavi SSH non vengono generate e l'utente non sarà in grado di accedere ai nodi di calcolo.

Il valore predefinito è `true`.

[Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.](#)

## AdditionalSssdConfigs(Facoltativo,) Dict

Un dizionario di coppie chiave-valore che contengono parametri e valori SSSD da scrivere nel file di configurazione SSSD sulle istanze del cluster. Per una descrizione completa del file di configurazione SSSD, consulta le pagine di manuale relative all'istanza e i relativi file di configurazione. SSSD

I parametri e i valori SSSD devono essere compatibili con AWS ParallelCluster la configurazione SSSD descritta nell'elenco seguente.

- `id_provider` è impostato su `ldap` internamente da AWS ParallelCluster e non deve essere modificato.
- `access_provider` è impostata su `ldap` internamente da AWS ParallelCluster quando [LdapAccessFilter](#) viene specificato [DirectoryService](#) e questa impostazione non deve essere modificata.

Se [DirectoryService](#)/[LdapAccessFilter](#) viene omissa, viene omissa anche la sua `access_provider` specificazione. Ad esempio, se si imposta su `access_provider` `simple` in [AdditionalSssdConfigs](#), non è [LdapAccessFilter](#) necessario specificare [DirectoryService](#)/.

I seguenti frammenti di configurazione sono esempi di configurazioni valide per `AdditionalSssdConfigs`

Questo esempio abilita il livello di debug per i log SSSD, limita la base di ricerca a un'unità organizzativa specifica e disabilita la memorizzazione nella cache delle credenziali.

```
DirectoryService:
...
AdditionalSssdConfigs:
 debug_level: "0xFFFF0"
 ldap_search_base: OU=Users,OU=CORP,DC=corp,DC=example,DC=com
 cache_credentials: False
```

Questo esempio specifica la configurazione di un SSD. [simple](#) `access_provider` Agli utenti di `EngineeringTeam` viene fornito l'accesso alla directory. [DirectoryService](#)/non [LdapAccessFilter](#) deve essere impostato in questo caso.

```
DirectoryService:
```

```
...
AdditionalSsdConfigs:
 access_provider: simple
 simple_allow_groups: EngineeringTeam
```

Politica di aggiornamento: la flotta di elaborazione deve essere interrotta affinché questa impostazione possa essere modificata per un aggiornamento.

## Sezione **DeploymentSettings**

### Note

DeploymentSettings viene aggiunto a partire dalla AWS ParallelCluster versione 3.4.0.

(Facoltativo) Specifica la configurazione delle impostazioni di distribuzione.

```
DeploymentSettings:
 LambdaFunctionsVpcConfig:
 SecurityGroupIds:
 - string
 SubnetIds:
 - string
 DisableSudoAccessForDefaultUser: Boolean
 DefaultUserHome: string # 'Shared' or 'Local'
```

## Proprietà **DeploymentSettings**

### **LambdaFunctionsVpcConfig**

(Facoltativo) Specificate le configurazioni VPC AWS Lambda delle funzioni. Per ulteriori informazioni, consulta [AWS Lambda Configurazione VPC in AWS ParallelCluster](#).

```
LambdaFunctionsVpcConfig:
 SecurityGroupIds:
 - string
 SubnetIds:
 - string
```

## LambdaFunctionsVpcConfig properties

SecurityGroupIds(Obbligatorio,) [String]

L'elenco dei gruppi di sicurezza Amazon VPC collegati alle IDs funzioni Lambda.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

SubnetIds(Obbligatorio,) [String]

L'elenco delle IDs sottoreti collegate alle funzioni Lambda.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

### Note

Le sottoreti e i gruppi di sicurezza devono trovarsi nello stesso VPC.

DisableSudoAccessForDefaultUser proprietà

### Note

Questa opzione di configurazione è supportata solo con i Slurm cluster.

(Facoltativo) Se `True`, i privilegi sudo dell'utente predefinito verranno disabilitati. Questo vale per tutti i nodi del cluster.

```
Main DeploymentSettings section in config yaml(applyes to HN, CF and LN)
DeploymentSettings:
 DisableSudoAccessForDefaultUser: True
```

Per aggiornare il valore di `DisableSudoAccessForDefaultUser`, è necessario interrompere la flotta di elaborazione e tutti i nodi di accesso.

Politica di aggiornamento: la flotta di elaborazione e i nodi di accesso devono essere interrotti affinché questa impostazione venga modificata per un aggiornamento.

## DefaultUserHome proprietà

Se impostato su `Shared`, il cluster utilizzerà la configurazione predefinita e condividerà la directory dell'utente predefinito in tutto il cluster entro `/home/<default user>`.

Se impostato su `Local`, il nodo principale, i nodi di accesso e i nodi di calcolo avranno ciascuno una directory utente locale predefinita separata in `local/home/<default user>`.

## Crea file di configurazione delle immagini

AWS ParallelCluster la versione 3 utilizza i file YAML 1.1 per i parametri di configurazione dell'immagine di compilazione. Conferma che l'indentazione sia corretta per ridurre gli errori di configurazione. Per ulteriori informazioni, consulta le specifiche YAML 1.1 all'indirizzo. <https://yaml.org/spec/1.1/>

Questi file di configurazione vengono utilizzati per definire il modo in cui le AWS ParallelCluster AMI personalizzate vengono create utilizzando EC2 Image Builder. I processi di creazione di AMI personalizzati vengono attivati utilizzando il [pcluster build-image](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/build-image) comando. Per alcuni esempi di file di configurazione, vedi [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test\\_imagebuilder\\_schema/test\\_imagebuilder\\_schema](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema).

### Argomenti

- [Crea le proprietà del file di configurazione dell'immagine](#)
- [Sezione Build](#)
- [Sezione Image](#)
- [Sezione DeploymentSettings](#)

## Crea le proprietà del file di configurazione dell'immagine

Region(Facoltativo,String)

Specifica il tipo Regione AWS di build-image operazione. Ad esempio, `us-east-2`.

CustomS3Bucket(Facoltativo,**String**)

Specifica il nome di un bucket Amazon S3 creato nel AWS tuo account per archiviare le risorse utilizzate dal processo di creazione dell'AMI personalizzata e per esportare i log. Le informazioni utilizzate dall'immagine si trovano nel bucket personalizzato per la configurazione dell'immagine.

AWS ParallelCluster mantiene un bucket Amazon S3 in ogni AWS regione in cui crei i cluster. Per

impostazione predefinita, questi bucket Amazon S3 sono denominati `parallelcluster-hash-v1-D0-NOT-DELETE`

## Sezione **Build**

(Obbligatorio) Specificate la configurazione in cui verrà creata l'immagine.

```
Build:
 Imds:
 ImdsSupport: string
 InstanceType: string
 SubnetId: string
 ParentImage: string
 Iam:
 InstanceRole: string
 InstanceProfile: string
 CleanupLambdaRole: string
 AdditionalIamPolicies:
 - Policy: string
 PermissionsBoundary: string
 Components:
 - Type: string
 Value: string
 Tags:
 - Key: string
 Value: string
 SecurityGroupIds:
 - string
 UpdateOsPackages:
 Enabled: boolean
 Installation:
 NvidiaSoftware:
 Enabled: boolean
 LustreClient:
 Enabled: boolean
```

## Proprietà **Build**

InstanceType(Obbligatorio,String)

Specifica il tipo di istanza utilizzata per creare l'immagine.

## SubnetId(Facoltativo,**String**)

Specifica l'ID di una sottorete esistente in cui effettuare il provisioning dell'istanza per creare l'immagine. La sottorete fornita richiede l'accesso a Internet. Tieni presente che potrebbe essere necessario [modificare gli attributi di indirizzo IP della sottorete se la](#) compilazione fallisce.

### Warning

`pcluster build-image` utilizza il VPC predefinito. Se il VPC predefinito è stato eliminato, magari utilizzando AWS Control Tower o AWS Landing Zone, è necessario specificare l'ID della sottorete.

Quando si specifica il `SubnetId`, si consiglia di specificare anche la `SecurityGroupIds` proprietà. Se ometti, AWS ParallelCluster utilizzerà `SecurityGroupIds` i gruppi di sicurezza predefiniti o si baserà sul comportamento predefinito all'interno della sottorete specificata. Quando si utilizzano entrambi, si ottengono i seguenti vantaggi:

- **Controllo granulare:** quando li definisci in modo esplicito entrambi, ti assicuri che le istanze avviate durante il processo di creazione dell'immagine siano collocate nella sottorete corretta e disponga dell'accesso preciso alla rete necessario per i componenti della build e tutti i servizi richiesti (come l'accesso a S3 per gli script di compilazione).
- **Migliori pratiche di sicurezza:** quando definisci i gruppi di sicurezza appropriati, ciò aiuta a limitare l'accesso alla rete solo alle porte e ai servizi necessari, il che migliora la sicurezza dell'ambiente di compilazione.
- **Evitare potenziali problemi:** se ci si affida esclusivamente ai valori predefiniti, è possibile che i gruppi di sicurezza siano troppo aperti o troppo restrittivi, il che può causare problemi durante il processo di creazione.

## ParentImage(Obbligatorio,) String

Specifica l'immagine di base. L'immagine principale può essere non AWS ParallelCluster AMI o un' AWS ParallelCluster AMI ufficiale per la stessa versione. Non puoi usare un'AMI AWS ParallelCluster ufficiale o personalizzata di una versione diversa di AWS ParallelCluster. Il formato deve essere l'ARN di un'immagine `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion` o un ID AMI. `ami-12345678`



## SecurityGroupIds(Facoltativo, **[String]**)

Specifica l'elenco dei gruppi di sicurezza IDs per l'immagine.

## Imds

### Proprietà **Imds**

(Facoltativo) Specifica le impostazioni IMDS (Build and Test Instance Metadata Service) di Amazon ImageBuilder EC2.

```
Imds:
ImdsSupport: string
```

### ImdsSupport(Facoltativo,) String

Specifica quali versioni IMDS sono supportate nelle istanze di build e test di Amazon ImageBuilder EC2. I valori supportati sono `v2.0` e `v1.0`. Il valore predefinito è `v2.0`.

Se `ImdsSupport` è impostato su `v1.0`, entrambi IMDSv1 sono supportati. IMDSv2

Se `ImdsSupport` è impostato su `v2.0`, solo IMDSv2 è supportato.

Per ulteriori informazioni, consulta [Use IMDSv2](#) in the Amazon EC2 User Guide for Linux instances.

Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.

#### Note

A partire dalla AWS ParallelCluster versione 3.7.0, il `ImdsSupport` valore predefinito è `v2.0`. Ti consigliamo di impostare `v2.0` e `ImdsSupport` sostituire IMDSv1 con IMDSv2 nelle tue azioni personalizzate le chiamate.

Il supporto per [Imds/ImdsSupport](#) viene aggiunto con la AWS ParallelCluster versione 3.3.0.

## Iam

### Proprietà Iam

(Facoltativo) Specifica le risorse IAM per la creazione dell'immagine.

```
Iam:
 InstanceRole: string
 InstanceProfile: string
 CleanupLambdaRole: string
 AdditionalIamPolicies:
 - Policy: string
 PermissionsBoundary: string
```

### InstanceProfile(Facoltativo,String)

Speciifica un profilo di istanza per sovrascrivere il profilo di istanza predefinito per l'istanza EC2 Image Builder. InstanceProfile e AdditionalIamPolicies non possono essere InstanceRole specificati insieme. Il formato è `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

### InstanceRole(Facoltativo,String)

Speciifica un ruolo di istanza per sovrascrivere il ruolo di istanza predefinito per l'istanza EC2 Image Builder. InstanceProfile e AdditionalIamPolicies non possono essere InstanceRole specificati insieme. Il formato è `arn:Partition:iam::Account:role/RoleName`.

### CleanupLambdaRole(Facoltativo,String)

L'ARN del ruolo IAM da utilizzare per la AWS Lambda funzione che supporta la risorsa CloudFormation personalizzata che rimuove gli artefatti della build al completamento della build. Lambda deve essere configurata come principale autorizzata ad assumere il ruolo. Il formato è `arn:Partition:iam::Account:role/RoleName`.

### AdditionalIamPolicies(Facoltativo)

Specifica policy IAM aggiuntive da collegare all'istanza EC2 Image Builder utilizzata per produrre l'AMI personalizzata.

```
AdditionalIamPolicies:
 - Policy: string
```

## Policy(Facoltativo,) [String]

Elenco delle politiche IAM. Il formato è

arn:*Partition*:iam::*Account*:policy/*PolicyName*.

## PermissionsBoundary(Facoltativo,String)

L'ARN della policy IAM da utilizzare come limite delle autorizzazioni per tutti i ruoli creati da AWS ParallelCluster. Per ulteriori informazioni sui limiti delle autorizzazioni IAM, consulta la sezione Limiti delle [autorizzazioni per le entità IAM nella Guida per l'utente IAM](#). Il formato è arn:*Partition*:iam::*Account*:policy/*PolicyName*.

## Components

### Proprietà **Components**

(Facoltativo) Specifica i componenti di Amazon ImageBuilder EC2 da utilizzare durante il processo di creazione dell'AMI oltre a quelli forniti di default da AWS ParallelCluster. Tali componenti possono essere utilizzati per personalizzare il processo di creazione dell'AMI. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

#### Components:

- Type: *string*
- Value: *string*

## Type(Facoltativo,String)

Specifica il tipo di coppia tipo-valore per il componente. Il tipo può essere o. arn script

## Value(Facoltativo,String)

Specifica il valore della coppia tipo-valore per il componente. Quando type è arn, questo è l'ARN di un componente EC2 Image Builder. Quando il tipo è script, questo è il percorso. https or s3 link that points to the script to use when you create the EC2 Image Builder com

## Tags

### Proprietà **Tags**

(Facoltativo) Specifica l'elenco di tag da impostare nelle risorse utilizzate per creare l'AMI.

**Tags:**

- **Key:** *string*  
**Value:** *string*

Key(Facoltativo,**String**)

Definisce il nome del tag.

Value(Facoltativo,String)

Definisce il valore del tag.

## UpdateOsPackages

Proprietà **UpdateOsPackages**

(Facoltativo) Specificate se il sistema operativo viene aggiornato prima di installare lo AWS ParallelCluster stack software.

**UpdateOsPackages:**

**Enabled:** *boolean*

Enabled(Facoltativo,) Boolean

Set true, il sistema operativo viene aggiornato e riavviato prima di installare il AWS ParallelCluster software. Il valore predefinito è false.

 **Note**

Quando UpdateOsPackages è abilitato, tutti i pacchetti del sistema operativo disponibili vengono aggiornati, incluso il kernel. In qualità di cliente, è tua responsabilità verificare che l'aggiornamento sia compatibile con le dipendenze AMI non incluse nell'aggiornamento.

Ad esempio, supponiamo di voler creare un'AMI per la AWS ParallelCluster versione X.0 fornita con la versione del kernel Y.0 e la versione di alcuni componenti Z.0. Supponiamo che l'aggiornamento disponibile includa la versione aggiornata del kernel Y.1 senza aggiornamenti al componente Z.0. Prima di abilitarlo UpdateOsPackages, è tua responsabilità verificare che il componente Z.0 supporti il kernel Y.1.

## Installation

### Proprietà **Installation**

(Facoltativo) Specifica il software aggiuntivo da installare sull'immagine.

```
Installation:
 NvidiaSoftware:
 Enabled: boolean
 LustreClient:
 Enabled: boolean
```

#### NvidiaSoftwareproprietà (facoltativo)

Specifica il software Nvidia da installare.

```
NvidiaSoftware:
 Enabled: boolean
```

Enabled(Facoltativo,) *boolean*

Settrue, verranno installati il driver GPU Nvidia e CUDA. Il valore predefinito è false.

#### LustreClientproprietà (opzionale)

Speciifica che verrà installato il client Amazon FSx Lustre.

```
LustreClient:
 Enabled: boolean
```

Enabled(Facoltativo,) *boolean*

Settrue, verrà installato il client Lustre. Il valore predefinito è true.

## Sezione **Image**

(Facoltativo) Definisce le proprietà dell'immagine per la creazione dell'immagine.

```
Image:
 Name: string
 RootVolume:
```

```
Size: integer
Encrypted: boolean
KmsKeyId: string
Tags:
- Key: string
 Value: string
```

## Proprietà **Image**

Name(Facoltativo,String)

Specificate il nome dell'AMI. Se non viene specificato, viene utilizzato il nome utilizzato per la chiamata del [pcluster build-image](#) comando.

## Tags

Proprietà **Tags**

(Facoltativo) Specificate le coppie chiave-valore per l'immagine.

```
Tags:
- Key: string
 Value: string
```

Key(Facoltativo,) String

Definisce il nome del tag.

Value(Facoltativo,String)

Definisce il valore del tag.

## RootVolume

Proprietà **RootVolume**

(Facoltativo) Specificate le proprietà del volume principale dell'immagine.

```
RootVolume:
 Size: integer
 Encrypted: boolean
```

KmsKeyId: *string*

### Size(Facoltativo,**Integer**)

Specifica la dimensione del volume root per l'immagine, in GiB. La dimensione predefinita è la dimensione dei [ParentImage](#) più 27 GiB.

### Encrypted(Facoltativo,**Boolean**)

Specifica se il volume è crittografato. Il valore predefinito è `false`.

### KmsKeyId(Facoltativo,**String**)

Specifica l'ARN della chiave utilizzata per crittografare AWS KMS il volume. Il formato è».

arn:*Partition*:kms:*Region*:*Account*:key/*KeyId*

## Sezione **DeploymentSettings**

### Note

`DeploymentSettings` viene aggiunto a partire dalla AWS ParallelCluster versione 3.4.0.

(Facoltativo) Specifica la configurazione delle impostazioni di distribuzione.

DeploymentSettings:

LambdaFunctionsVpcConfig:

SecurityGroupIds:

- *string*

SubnetIds:

- *string*

## Proprietà **DeploymentSettings**

### **LambdaFunctionsVpcConfig**

(Facoltativo) Specificate le configurazioni VPC AWS Lambda delle funzioni. Per ulteriori informazioni, consulta [AWS Lambda Configurazione VPC in AWS ParallelCluster](#).

LambdaFunctionsVpcConfig:

SecurityGroupIds:

```
- string
SubnetIds:
- string
```

## LambdaFunctionsVpcConfig properties

SecurityGroupIds(Obbligatorio,) [String]

L'elenco dei gruppi di sicurezza Amazon VPC collegati alle IDs funzioni Lambda.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

SubnetIds(Obbligatorio,) [String]

L'elenco delle IDs sottoreti collegate alle funzioni Lambda.

[Politica di aggiornamento: se questa impostazione viene modificata, l'aggiornamento non è consentito.](#)

### Note

Le sottoreti e i gruppi di sicurezza devono trovarsi nello stesso VPC.

## AWS ParallelCluster Riferimento API

Questa sezione fornisce descrizioni, sintassi ed esempi di utilizzo per ciascuna delle azioni AWS ParallelCluster API.

### Argomenti

- [BuildImage](#)
- [createCluster](#)
- [Elimina cluster](#)
- [deleteClusterInstances](#)
- [Elimina immagine](#)
- [Descrivi Cluster](#)
- [describeClusterInstances](#)



- [describeComputeFleet](#)
- [Descrivi l'immagine](#)
- [getClusterLogEventi](#)
- [getClusterStackEventi](#)
- [getImageLogEventi](#)
- [getImageStackEventi](#)
- [Elenca i cluster](#)
- [listClusterLogStream](#)
- [listImageLogStream](#)
- [Elenca immagini](#)
- [listOfficialImages](#)
- [Aggiorna Cluster](#)
- [updateComputeFleet](#)

## BuildImage

Crea un' AWS ParallelCluster immagine personalizzata in un file. Regione AWS

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
POST /v3/images/custom
{
 "imageConfiguration": "string",
 "imageId": "string",
 "dryrun": boolean,
 "region": "string",
```

```
"rollbackOnFailure": boolean,
"supressValidators": ["string"],
"validationFailureLevel": "string"
}
```

## Corpo di richiesta

### Configurazione dell'immagine

La configurazione dell'immagine come documento YAML.

Tipo: stringa

Campo obbligatorio: sì

imageId

L'ID dell'immagine da creare.

Tipo: stringa

Campo obbligatorio: sì

funzionamento a secco

Se impostato su `true`, esegue solo la convalida della richiesta senza creare alcuna risorsa.

Utilizzate questo parametro per convalidare la configurazione dell'immagine. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

### Regione

Il modo Regione AWS in cui si esegue il comando per creare l'immagine.

Tipo: stringa

Campo obbligatorio: no

rollbackOnFailure

Se impostato su `true`, il rollback dello stack di immagini si verifica se l'immagine non viene creata. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

Sopprimere i validatori

Identifica uno o più validatori di configurazione da sopprimere.

Tipo: elenco di stringhe

Formato: (ALL | type: [A-Za-z0-9]+)

Campo obbligatorio: no

validationFailureLevel

Il livello di convalida minimo che causa il fallimento della creazione dell'immagine. Il valore predefinito è ERROR.

Tipo: stringa

Valori validi: INFO | WARNING | ERROR

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "image": {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS"
 },
 "validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
```

```
 "message": "string"
 }
]
}
```

## Corpo di risposta

### image

#### imageId

L'ID dell'immagine.

Tipo: stringa

#### cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

#### cloudformationStackStatus

Lo stato dello CloudFormation stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

#### ec2 AmiInfo

##### ami\_id

L'ID Amazon EC2 AMI.

Tipo: stringa

#### imageBuildStatus

Lo stato di creazione dell'immagine.

Tipo: stringa

Valori validi: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

Regione

Lo Regione AWS in cui è costruita l'immagine.

Tipo: stringa

version

La AWS ParallelCluster versione utilizzata per creare l'immagine.

Tipo: stringa

Messaggi di convalida

Un elenco di messaggi con un livello di convalida inferiore a `validationFailureLevel`.  
L'elenco dei messaggi viene raccolto durante la convalida della configurazione.

id

L'ID del validatore.

Tipo: stringa

level

Il livello di convalida.

Tipo: stringa

Valori validi: INFO | WARNING | ERROR

message

Un messaggio di convalida.

Tipo: stringa

tipo

Il tipo di validatore.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ build_image(custom-image-id, custom-image-config.yaml)
```

#### 200 Risposta

```
{
 "image": {
 "cloudformation_stack_arn": "arn:aws:cloudformation:us-
east-1:123456789012:stack/custom-image-id/711b76b0-af81-11ec-a29f-0ee549109f1f",
 "cloudformation_stack_status": "CREATE_IN_PROGRESS",
 "image_build_status": "BUILD_IN_PROGRESS",
 "image_id": "custom-image-id",
 "region": "us-east-1",
 "version": "3.2.1"
 }
}
```

## createCluster

Crea un cluster gestito in un Regione AWS.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
POST /v3/clusters
{
 "clusterName": "string",
```

```
"clusterConfiguration": "string",
"dryrun": boolean,
"region": "string",
"rollbackOnFailure", boolean,
"suppressValidators": ["string"],
"validationFailureLevel": "string"
}
```

## Corpo di richiesta

### Configurazione del cluster

La configurazione del cluster come documento YAML.

Tipo: stringa

Campo obbligatorio: sì

### ClusterName

Il nome del cluster da creare.

Il nome deve iniziare con un carattere alfabetico. Il nome può contenere fino a 60 caratteri. Se la Slurm contabilità è abilitata, il nome può contenere fino a 40 caratteri.

Tipo: stringa

Campo obbligatorio: sì

### funzionamento a secco

Se impostato su `true`, esegue solo la convalida della richiesta ma non crea alcuna risorsa.

Utilizzate questo parametro per convalidare la configurazione del cluster. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

### Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

rollbackOnFailure

Se impostato su `true`, il rollback dello stack del cluster si verifica se il cluster non riesce a creare. Il valore predefinito è `true`.

Tipo: booleano

Campo obbligatorio: no

Sopprimere i validatori

Identifica uno o più validatori di configurazione da sopprimere.

Tipo: elenco di stringhe

Formato: (ALL | type: [A-Za-z0-9]+)

Campo obbligatorio: no

validationFailureLevel

Il livello di convalida minimo che causa l'esito negativo della creazione del cluster. Il valore predefinito è `ERROR`.

Tipo: stringa

Valori validi: INFO | WARNING | ERROR

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
```



```
 "metadata": {
 "name": "string",
 "version": "string"
 }
 },
 "validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
]
}
```

## Corpo di risposta

### ClusterName

Il nome del cluster.

Tipo: stringa

### cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

### cloudformationStackStatus

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

### Stato del cluster

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## Regione

Il in Regione AWS cui viene creato il cluster.

Tipo: stringa

## pianificatore

### metadata

I metadati dello scheduler

#### nome

Il nome dello scheduler.

Tipo: stringa

#### version

La versione dello scheduler.

Tipo: stringa

## tipo

Il tipo di scheduler.

Tipo: stringa

## version

La AWS ParallelCluster versione utilizzata per creare il cluster.

Tipo: stringa

## validation\_messages

Un elenco di messaggi con un livello di convalida inferiore a. `validationFailureLevel`  
L'elenco dei messaggi viene raccolto durante la convalida della configurazione.

### id

L'ID del validatore.

Tipo: stringa

level

Tipo: stringa

Valori validi: INFO | WARNING | ERROR

message

Un messaggio di convalida.

Tipo: stringa

tipo

Il tipo di validatore.

Tipo: stringa

## Esempio

### Python

Richiesta

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

200 Risposta

```
{
 "cluster": {
 "cloudformation_stack_arn": "arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster-3x/e0462730-50b5-11ed-99a3-0a5ddc4a34c7",
 "cloudformation_stack_status": "CREATE_IN_PROGRESS",
 "cluster_name": "cluster-3x",
 "cluster_status": "CREATE_IN_PROGRESS",
 "region": "us-east-1",
 "scheduler": {
 "type": "slurm"
 },
 "version": "3.2.1"
 }
}
```

# Elimina cluster

Inizia l'eliminazione di un cluster.

## Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
DELETE /v3/clusters/{clusterName}
{
 "region": "string"
}
```

## Corpo di richiesta

### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

### Regione

Il file Regione AWS in cui viene eliminato il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
```

```
"cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "clusterStatus": "DELETE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
}
```

## Corpo di risposta

### cluster

Un elenco di istanze del cluster.

#### ClusterName

Il nome di un cluster.

Tipo: stringa

#### cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

#### cloudformationStackStatus

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |

UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

Stato del cluster

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

Regione

Regione AWS In cui viene creato il cluster.

Tipo: stringa

pianificatore

metadata

I metadati dello scheduler.

nome

Il nome dello scheduler.

Tipo: stringa

version

La versione dello scheduler.

Tipo: stringa

tipo

Il tipo di scheduler.

Tipo: stringa

version

La AWS ParallelCluster versione utilizzata per creare il cluster.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ delete_cluster(cluster_name_3x)
```

#### 200 Risposta

```
{
 "cluster": {
 "cloudformation_stack_arn": "arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-aee5-11ec-8e18-0ac1d712b241",
 "cloudformation_stack_status": "DELETE_IN_PROGRESS",
 "cluster_name": "cluster_name_3x",
 "cluster_status": "DELETE_IN_PROGRESS",
 "region": "us-east-1",
 "version": "3.2.1"
 }
}
```

## deleteClusterInstances

Avvia la chiusura forzata di tutti i nodi di calcolo del cluster. Questa azione non supporta i cluster.

### AWS Batch

#### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
DELETE /v3/clusters/{clusterName}/instances
{
 "force": boolean,
```

```
"region": "string"
}
```

## Corpo di richiesta

### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

### force

Se impostato su `true`, forza l'eliminazione quando il cluster con il nome specificato non viene trovato. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

### Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Corpo di risposta

Nessuno

## Esempio

### Python

Richiesta

```
$ delete_cluster_instances(cluster_name_3x)
```

200 Risposta



Nessuno

## Elimina immagine

Avvia l'eliminazione dell'immagine personalizzata. AWS ParallelCluster

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
DELETE /v3/images/custom/{imageId}
{
 "force": boolean,
 "region": "string"
}
```

### Corpo di richiesta

#### imageId

L'ID dell'immagine.

Tipo: stringa

Campo obbligatorio: sì

#### force

Se impostato su `true`, forza l'eliminazione dell'AMI. Utilizza questo parametro se esistono istanze che utilizzano l'AMI o se l'AMI è condiviso. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

## Regione

Il formato Regione AWS in cui è stata creata l'immagine.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "image": {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS"
 }
}
```

## Corpo di risposta

### image

#### cloudformationStackArn

Il nome della risorsa Amazon (ARN) dello stack principale CloudFormation .

Tipo: stringa

#### cloudformationStackStatus

Lo stato dello CloudFormation stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |

UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

ec2 AmiInfo

ID AMI

L'ID Amazon EC2 AMI.

Tipo: stringa

imageBuildStatus

Lo stato di creazione dell'immagine.

Tipo: stringa

Valori validi: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

imageId

L'ID dell'immagine.

Tipo: stringa

Regione

L' Regione AWS elemento in cui viene creata l'immagine.

Tipo: stringa

version

La AWS ParallelCluster versione utilizzata per creare l'immagine.

Tipo: stringa

## Esempio

Python

Richiesta

```
$ delete_image(custom-image-id)
```

## 200 Risposta

```
{
 "image": {
 "image_build_status": "DELETE_IN_PROGRESS",
 "image_id": "custom-image-id",
 "region": "us-east-1",
 "version": "3.2.1"
 }
}
```

## Descrivi Cluster

Ottieni informazioni dettagliate su un cluster esistente.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /v3/clusters/{clusterName}
{
 "region": "string"
}
```

### Corpo di richiesta

#### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

## Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

### Note

`failureReason` è passato alla versione 3.5.0 `failures` a partire dalla AWS ParallelCluster versione 3.5.0.

```
{
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudFormationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 },
 "cloudFormationStackArn": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "lastUpdatedTime": "2019-08-24T14:15:22Z",
 "clusterConfiguration": {
 "url": "string"
 },
 "computeFleetStatus": "START_REQUESTED",
 "tags": [
 {
```

```
 "key": "string",
 "value": "string"
 }
],
"headNode": {
 "instanceId": "string",
 "instanceType": "string",
 "launchTime": "2019-08-24T14:15:22Z",
 "privateIpAddress": "string",
 "publicIpAddress": "string",
 "state": "pending"
},
"failures": [
 {
 "failureCode": "string",
 "failureReason": "string"
 }
],
"loginNodes": {
 "status": "string",
 "address": "string",
 "poolName": "string",
 "scheme": "string",
 "healthyNodes": integer,
 "unhealthyNodes": integer
}
}
```

## Corpo di risposta

### ClusterName

Il nome del cluster.

Tipo: stringa

### cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

### cloudformationStackStatus

Lo stato dello CloudFormation stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

Configurazione del cluster

url

L'URL del file di configurazione del cluster.

Tipo: stringa

Stato del cluster

Lo stato del cluster.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

computeFleetStatus

Lo stato della flotta di elaborazione.

Tipo: stringa

Valori validi: START\_REQUESTED | STARTING | RUNNING | PROTECTED |  
STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

creationTime

Timestamp di quando è stato creato il cluster.

Tipo: datetime

lastUpdatedTime

Data/ora dell'ultimo aggiornamento del cluster.

Tipo: datetime

Regione

Il luogo Regione AWS in cui viene creato il cluster.

Tipo: stringa

tags

L'elenco dei tag associati al cluster.

Chiave

Nome del tag.

Tipo: stringa

tag

Valore del tag.

Tipo: stringa

version

La AWS ParallelCluster versione utilizzata per creare il cluster.

Tipo: stringa

failures

L'elenco degli errori quando lo stack del cluster è in CREATE\_FAILED stato.

Codice di errore

Il codice di errore quando lo stack del cluster è in stato. CREATE\_FAILED

Tipo: stringa

Motivo dell'errore

Il motivo dell'errore quando lo stack del cluster è in stato. CREATE\_FAILED

Tipo: stringa

head\_node

Il nodo principale del cluster.



## instanceId

L'ID dell' EC2 istanza Amazon.

Tipo: stringa

## instanceType

Il tipo di EC2 istanza Amazon.

Tipo: stringa

## launchTime

L'ora in cui è stata lanciata l' EC2 istanza Amazon.

Tipo: datetime

## privateIpAddress

L'indirizzo IP privato del cluster.

Tipo: stringa

## publicIpAddress

L'indirizzo IP pubblico del cluster.

Tipo: stringa

## stato

Lo stato dell'istanza del nodo principale.

Tipo: stringa

Valori validi: pending | running | shutting-down | terminated | stopping | stopped

## pianificatore

### metadata

I metadati dello scheduler.

### nome

Il nome dello scheduler.

Tipo: stringa

version

La versione dello scheduler.

Tipo: stringa

Nodi di accesso

stato

Lo stato del nodo di accesso.

Tipo: stringa

Valori validi: PENDING | FAILED | ACTIVE

address

L'indirizzo del nodo di accesso.

Tipo: stringa

Nome del pool

Il nome del pool del nodo di accesso.

Tipo: stringa

scheme

Lo schema dei nodi di accesso.

Tipo: stringa

Nodi sani

Il numero di nodi sani.

Tipo: Integer

Nodi non sani

Il numero di nodi non sani.

Tipo: Integer

## tipo

Il tipo di scheduler.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ describe_cluster(cluster_name_3x)
```

#### 200 Risposta

```
{
 "cloud_formation_stack_status": "CREATE_COMPLETE",
 "cloudformation_stack_arn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241",
 "cluster_configuration": {
 "url": "https://parallelcluster-...."
 },
 "cluster_name": "cluster_name_3x",
 "cluster_status": "CREATE_COMPLETE",
 "compute_fleet_status": "RUNNING",
 "creation_time": datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
 "head_node": {
 "instance_id": "i-abcdef01234567890",
 "instance_type": "t2.micro",
 "launch_time": datetime.datetime(2022, 3, 28, 22, 21, 56, tzinfo=tzlocal()),
 "private_ip_address": "172.31.56.3",
 "public_ip_address": "107.23.100.164",
 "state": "running"
 },
 "last_updated_time": datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
 "region": "us-east-1",
 "tags": [
 {
 "key": "parallelcluster:version", "value": "3.2.1"
 }
]
}
```

```
 }
],
 "version": "3.2.1"
}
```

## describeClusterInstances

Descrivi le istanze che appartengono a un cluster.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /v3/clusters/{clusterName}/instances
{
 "nextToken": "string",
 "nodeType": "string",
 "queueName": "string",
 "region": "string"
}
```

### Corpo di richiesta

#### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

#### nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

## Tipo di nodo

Filtra le istanze per tipo di nodo.

Tipo: stringa

Valori validi: HeadNode, ComputeNode, LoginNode

Campo obbligatorio: no

## QueueName

Filtra le istanze per nome della coda.

Tipo: stringa

Campo obbligatorio: no

## Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "instances": [
 {
 "instanceId": "string",
 "instanceType": "string",
 "launchTime": "2019-08-24T14:15:22Z",
 "privateIpAddress": "string",
 "publicIpAddress": "string",
 "state": "pending",
 "nodeType": "HeadNode",
 "queueName": "string",

```

```
 "poolName": "string"
 }
]
}
```

## Corpo di risposta

### Istanze

L'elenco delle istanze del cluster.

`instanceId`

L'ID dell' EC2 istanza Amazon.

Tipo: stringa

`instanceType`

Il tipo di EC2 istanza Amazon.

Tipo: stringa

`launchTime`

L'ora in cui è stata lanciata l' EC2 istanza Amazon.

Tipo: datetime

`Tipo di nodo`

Il tipo di nodo.

Tipo: stringa

Valori validi: `HeadNode`, `ComputeNode`, `LoginNode`

`publicIpAddress`

L'indirizzo IP pubblico del cluster.

Tipo: stringa

`QueueName`

Il nome della coda in cui l' EC2 istanza Amazon esegue il backup di un nodo.

Tipo: stringa

stato

Lo stato dell' EC2 istanza Amazon del nodo.

Tipo: stringa

Valori validi: pending | running | shutting-down | terminated | stopping | stopped

nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure null se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

Python

Richiesta

```
$ describe_cluster_instances(cluster_name_3x)
```

200 Risposta

```
{
 "instances": [
 {
 "instance_id": "i-abcdef01234567890",
 "instance_type": "t2.micro",
 "launch_time": datetime.datetime(2022, 3, 30, 14, 2, 7, tzinfo=tzlocal()),
 "node_type": "HeadNode",
 "private_ip_address": "192.0.2.5",
 "public_ip_address": "198.51.100.180",
 "state": "running"
 }
]
}
```

# describeComputeFleet

Descrivi lo stato della flotta di elaborazione.

## Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
GET /v3/clusters/{clusterName}/computefleet
{
 "region": "string"
}
```

## Corpo di richiesta

### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

### Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
```



```
"status": "START_REQUESTED",
"lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

## Corpo di risposta

stato

Tipo: stringa

Valori validi: START\_REQUESTED | STARTING | RUNNING | PROTECTED |  
STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

lastStatusUpdatedL'ora

Il timestamp che rappresenta l'ora dell'ultimo aggiornamento dello stato.

Tipo: datetime

## Esempio

Python

Richiesta

```
$ describe_compute_fleet(cluster_name_3x)
```

200 Risposta

```
{
 "last_status_updated_time": datetime.datetime(2022, 3, 28, 22, 27, 14,
 tzinfo=tzlocal()),
 "status": "RUNNING"
}
```

## Descrivi l'immagine

Ottieni informazioni dettagliate su un'immagine esistente.

Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
GET /v3/images/custom/{imageId}
{
 "region": "string"
}
```

### Corpo di richiesta

#### imageId

L'ID dell'immagine.

Tipo: stringa

Campo obbligatorio: sì

#### Regione

L' Regione AWS elemento in cui è stata creata l'immagine.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "imageId": "string",
 "region": "string",
 "version": "string",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "imageBuildLogsArn": "string",
```

```
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackStatusReason": "string",
"cloudformationStackArn": "string",
"creationTime": "2019-08-24T14:15:22Z",
"cloudformationStackCreationTime": "2019-08-24T14:15:22Z",
"cloudformationStackTags": [
 {
 "key": "string",
 "value": "string"
 }
],
"imageConfiguration": {
 "url": "string"
},
"imagebuilderImageStatus": "PENDING",
"imagebuilderImageStatusReason": "string",
"ec2AmiInfo": {
 "amiId": "string",
 "tags": [
 {
 "key": "string",
 "value": "string"
 }
],
 "amiName": "string",
 "architecture": "string",
 "state": "PENDING",
 "description": "string"
}
}
```

## Corpo di risposta

### imageId

L'ID dell'immagine per cui recuperare informazioni dettagliate.

Tipo: stringa

### imageBuildStatus

Lo stato di creazione dell'immagine.

Tipo: stringa

Valori validi: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

## Configurazione dell'immagine

url

L'URL del file di configurazione dell'immagine.

Tipo: stringa

## Regione

Il Regione AWS file in cui viene creata l'immagine.

Tipo: stringa

version

La AWS ParallelCluster versione utilizzata per creare l'immagine.

Tipo: stringa

cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

cloudformationStackCreationOra

L'ora in cui è stato creato CloudFormation lo stack.

Tipo: datetime

cloudformationStackStatus

Lo CloudFormation stato dello stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## cloudformationStackStatusMotivo

Il motivo dello stato dello CloudFormation stack.

Tipo: stringa

## cloudformationStackTags

L'elenco dei tag per lo CloudFormation stack.

Chiave

Il nome del tag.

Tipo: stringa

value

Il valore del tag.

Tipo: stringa

## creationTime

Gli orari in cui l'immagine è stata creata.

Tipo: datetime

## ec2 AmiInfo

ID AM

L'ID Amazon EC2 AMI.

Tipo: stringa

Un nome

Il nome Amazon EC2 AMI.

Tipo: stringa

architecture

L'architettura Amazon EC2 AMI.

Tipo: stringa

stato

Lo stato dell' EC2 AMI Amazon.

Tipo: stringa

Valori validi: PENDING | AVAILABLE | INVALID | DEREGISTERED | TRANSIENT | FAILED | ERROR

tags

Elenco di tag Amazon EC2 AML.

Chiave

Nome del tag.

Tipo: stringa

value

Valore del tag.

Tipo: stringa

imagebuilderImageStatus

Lo stato ImageBuilder dell'immagine.

Tipo: stringa

Valori validi: PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING | INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

imagebuilderImageStatusMotivo

Il motivo per cui l' ImageBuilder immagine ha questo status.

Tipo: stringa

imageBuildLogsArn

L'Amazon Resource Name (ARN) dei log per il processo di creazione dell'immagine.

Tipo: stringa

## Esempio

Python

Richiesta

```
$ describe_image(custom-image-id)
```

## 200 Risposta

```
{
 "cloudformation_stack_arn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-image-id/6accc570-b080-11ec-845e-0e2dc6386985",
 "cloudformation_stack_creation_time": datetime.datetime(2022, 3, 30, 23, 23, 33,
731000, tzinfo=tzlocal()),
 "cloudformation_stack_status": "CREATE_IN_PROGRESS",
 "cloudformation_stack_tags": [
 {
 "key": "parallelcluster:version", "value": "3.2.1"
 },
 {
 "key": "parallelcluster:image_name",
 "value": 'custom-image-id'
 },
 {
 "key": "parallelcluster:custom-image-id",
 "value": "custom-image-id"
 },
 {
 "key": 'parallelcluster:amzn-s3-demo-bucket',
 "value": 'amzn-s3-demo-bucket'
 },
 {
 "key": "parallelcluster:s3_image_dir",
 "value": "parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0"
 },
 {
 "key": "parallelcluster:build_log",
 "value": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-image-id"
 },
 {
 "key": "parallelcluster:build_config",
 "value": "s3://amzn-s3-demo-bucket/parallelcluster/3.2.1/images/custom-image-
id-1234567890abcdef0/configs/image-config.yaml"
 }
],
 "image_build_logs_arn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image",
```

```
"image_build_status": "BUILD_IN_PROGRESS",
"image_configuration": {
 "url": "https://amzn-s3-demo-bucket.s3.amazonaws.com/parallelcluster/3.2.1/
images/custom-image-id-1234567890abcdef0/configs/image-config.yaml?..."
},
"image_id": 'custom-image-id',
"imagebuilder_image_status": "PENDING",
"region": "us-east-1",
"version": "3.2.1"
}
```

## getClusterLogEventi

Recupera gli eventi associati a un flusso di log.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /v3/clusters/{clusterName}/logstreams/{logStreamName}
{
 "endTime": datetime,
 "limit": float,
 "nextToken": "string",
 "region": "string",
 "startFromHead": boolean,
 "startTime": datetime
}
```

### Corpo di richiesta

#### ClusterName

Il nome del cluster.



Tipo: stringa

Campo obbligatorio: sì

#### logStreamName

Il nome del flusso di log.

Tipo: stringa

Campo obbligatorio: sì

#### endTime

La fine dell'intervallo di tempo, espressa in formato ISO 8601. Gli eventi con un timestamp uguale o successivo a questo orario non sono inclusi.

Tipo: datetime

Formato: 2021-01-01T20:00:00Z

Campo obbligatorio: no

#### limit

Il numero massimo di eventi di registro restituiti. Se non si specifica un valore, il numero massimo è il numero di eventi di registro che possono rientrare in una dimensione di risposta di 1 MB o fino a 10.000 eventi di registro.

Tipo: Float

Campo obbligatorio: no

#### nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

#### Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## startFromHead

Se impostato su `true`, vengono restituiti per primi i primi eventi di registro. Se il valore è `false`, gli ultimi eventi di registro vengono restituiti per primi. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

## startTime

L'inizio dell'intervallo di tempo, espresso nel formato ISO 8601. Sono inclusi gli eventi con un timestamp uguale a questo orario o successivo a tale orario.

Tipo: datetime

Formato: `2021-01-01T20:00:00Z`

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "prevToken": "string",
 "events": [
 {
 "timestamp": "2019-08-24T14:15:22Z",
 "message": "string"
 }
]
}
```

## Corpo di risposta

### eventi

Elenco di eventi filtrati.

### message

Il messaggio dell'evento.

Tipo: stringa

## timestamp

Il timestamp dell'evento.

Tipo: datetime

## nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

## PrevToken

Un token che può essere utilizzato per recuperare il set di risultati precedente o `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ get_cluster_log_events(cluster_name_3x, log_stream_name=ip-192-0-2-26.i-
abcdef01234567890.cfn-init)
```

#### 200 Risposta

```
"events": [
 {
 "message": "2022-09-22 16:40:15,127 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.us-east-1.amazonaws.com",
 "timestamp": "2022-09-22T16:40:15.127Z"
 },
 {
 "message": "2022-09-22 16:40:15,127 [DEBUG] Describing resource
HeadNodeLaunchTemplate in stack cluster_name_3x",
 "timestamp": "2022-09-22T16:40:15.127Z"
 },
 ...
]
```

]

## getClusterStackEventi

Recupera gli eventi associati allo stack di un cluster.

### Note

A partire dalla versione 3.6.0, AWS ParallelCluster utilizza gli stack annidati per creare le risorse associate alle code e alle risorse di calcolo. L'GetClusterStackEventsAPI e il `pcluster get-cluster-stack-events` comando restituiscono solo gli eventi dello stack principale del cluster. È possibile visualizzare gli eventi dello stack del cluster, inclusi quelli relativi alle code e alle risorse di calcolo, nella console. CloudFormation

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /v3/clusters/{clusterName}/stackevents
{
 "nextToken": "string",
 "region": "string"
}
```

### Corpo di richiesta

#### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "events": [
 {
 "stackId": "string",
 "eventId": "string",
 "stackName": "string",
 "logicalResourceId": "string",
 "physicalResourceId": "string",
 "resourceType": "string",
 "timestamp": "2019-08-24T14:15:22Z",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "string",
 "resourceProperties": "string",
 "clientRequestToken": "string"
 }
]
}
```

## Corpo di risposta

eventi

Elenco degli eventi filtrati.

## clientRequestToken

Il token passato all'azione che ha generato questo evento.

Tipo: stringa

## eventId

L'ID univoco di questo evento.

Tipo: stringa

## logicalResourceId

Il nome logico della risorsa specificata nel modello.

Tipo: stringa

## physicalResourceId

Il nome o l'identificatore univoco associato all'istanza fisica della risorsa.

Tipo: stringa

## Proprietà della risorsa

Un BLOB delle proprietà utilizzate per creare la risorsa.

Tipo: stringa

## Stato delle risorse

Lo stato della risorsa.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED  
| UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED  
| IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS |  
IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

## resourceStatusReason

Un messaggio di successo o di fallimento associato alla risorsa.

Tipo: stringa

resourceType

Il tipo di risorsa.

Tipo: stringa

StackID

Il nome ID univoco dell'istanza dello stack.

Tipo: stringa

StackName

Il nome associato a uno stack.

Tipo: stringa

timestamp

L'ora in cui lo stato è stato aggiornato.

Tipo: datetime

nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

Python

Richiesta

```
$ get_cluster_stack_events(cluster_name_3x)
```

200 Risposta

```
{
```

```
"events": [
 {
 "event_id": "590b3820-b081-11ec-985e-0a7af5751497",
 "logical_resource_id": "cluster_name_3x",
 "physical_resource_id": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9",
 "resource_status": "CREATE_COMPLETE",
 "resource_type": "AWS::CloudFormation::Stack",
 "stack_id": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9",
 "stack_name": "cluster_name_3x",
 "timestamp": datetime.datetime(2022, 3, 30, 23, 30, 13, 268000,
tzinfo=tzlocal())
 },
 ...
]
}
```

## getImageLogEventi

Recupera gli eventi associati a una creazione di immagini.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /v3/images/custom/{imageId}/logstreams/{logStreamName}
{
 "endTime": datetime,
 "limit": float,
 "nextToken": "string",
 "region": "string",
 "startFromHead": boolean,
 "startTime": datetime
```



```
}
```

## Corpo di richiesta

### imageId

L'ID dell'immagine.

Tipo: stringa

Campo obbligatorio: sì

### logStreamName

Il nome del logstream.

Tipo: stringa

Campo obbligatorio: sì

### endTime

La fine dell'intervallo di tempo, espressa nel formato ISO 8601. Gli eventi con un timestamp uguale o successivo a questo orario non sono inclusi.

Tipo: datetime

Formato: 2021-01-01T20:00:00Z

Campo obbligatorio: no

### limit

Il numero massimo di eventi di registro restituiti. Se non si specifica un valore, il numero massimo di eventi di registro è il numero massimo di eventi di registro che può contenere una dimensione di risposta di 1 MB, fino a 10.000 eventi di registro.

Tipo: Float

Campo obbligatorio: no

### nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

## Regione

Regione AWS Quello in cui è inserita l'immagine.

Tipo: stringa

Campo obbligatorio: no

## startFromHead

Se impostato su `true`, restituisce per primi i primi eventi di registro. Se impostato su `false`, restituisce prima gli ultimi eventi di registro. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

## startTime

L'inizio dell'intervallo di tempo, espresso nel formato ISO 8601. Sono inclusi gli eventi con un timestamp uguale a questo orario o successivo a tale orario.

Tipo: datetime

Formato: `2021-01-01T20:00:00Z`

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "prevToken": "string",
 "events": [
 {
 "timestamp": "2019-08-24T14:15:22Z",
 "message": "string"
 }
]
}
```

## Corpo di risposta

### eventi

Un elenco di eventi filtrati.

#### message

Il messaggio dell'evento.

Tipo: stringa

#### timestamp

Il timestamp dell'evento.

Tipo: datetime

### nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

### PrevToken

Un token che può essere utilizzato per recuperare il set di risultati precedente o `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

#### 200 Risposta

```
"events": [
 {
```

```
[{"message": "ExecuteBash: STARTED EXECUTION",
 "timestamp": "2022-04-05T15:51:20.228Z"},
{
 "message": "ExecuteBash: Created temporary directory: /tmp/1234567890abcdef0",
 "timestamp": "2022-04-05T15:51:20.228Z"},
...
]
```

## getImageStackEventi

Recupera gli eventi associati allo stack per la creazione di un'immagine.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /v3/images/custom/{imageId}/stackevents
{
 "nextToken": "string",
 "region": "string"
}
```

### Corpo di richiesta

#### imageId

L'ID dell'immagine.

Tipo: stringa

Campo obbligatorio: sì

## nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

## Regione

Regione AWS Quello in cui si trova l'immagine.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "events": [
 {
 "stackId": "string",
 "eventId": "string",
 "stackName": "string",
 "logicalResourceId": "string",
 "physicalResourceId": "string",
 "resourceType": "string",
 "timestamp": "2019-08-24T14:15:22Z",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "string",
 "resourceProperties": "string",
 "clientRequestToken": "string"
 }
]
}
```

## Corpo di risposta

### eventi

Un elenco di eventi filtrati.

## clientRequestToken

Il token passato all'azione che ha generato questo evento.

Tipo: stringa

## eventId

L'ID univoco di questo evento.

Tipo: stringa

## logicalResourceId

Il nome logico della risorsa specificata nel modello.

Tipo: stringa

## physicalResourceId

Il nome o l'identificatore univoco associato all'istanza fisica della risorsa.

Tipo: stringa

## Proprietà della risorsa

Un BLOB delle proprietà utilizzate per creare la risorsa.

Tipo: stringa

## Stato delle risorse

Lo stato della risorsa.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED | UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED | IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS | IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

## resourceStatusReason

Un messaggio di successo o di fallimento associato alla risorsa.

Tipo: stringa

## resourceType

Il tipo di risorsa.

Tipo: stringa

## StackID

Il nome ID univoco dell'istanza dello stack.

Tipo: stringa

## StackName

Il nome associato a uno stack.

Tipo: stringa

## timestamp

L'ora in cui lo stato è stato aggiornato.

Tipo: datetime

## nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure null se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ get_image_stack_events(image_id)
```

#### 200 Risposta

```
{
 'events': [
 {
```

```

 'event_id': 'ParallelClusterImage-
CREATE_IN_PROGRESS-2022-03-30T23:26:33.499Z',
 'logical_resource_id': 'ParallelClusterImage',
 'physical_resource_id': 'arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-alinux2-image/3.2.1/1',
 'resource_properties': {
 "InfrastructureConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
 "ImageRecipeArn": "arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-alinux2-image/3.2.1",
 "DistributionConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
 "EnhancedImageMetadataEnabled": "false",
 "Tags": {
 "parallelcluster:image_name": "alinux2-
image", "parallelcluster:image_id": "alinux2-image"
 }
 },
 'resource_status': 'CREATE_IN_PROGRESS',
 'resource_status_reason': 'Resource creation Initiated',
 'resource_type': 'AWS::ImageBuilder::Image',
 'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/alinux2-
image/6accc570-b080-11ec-845e-0e2dc6386985',
 'stack_name': 'alinux2-image',
 'timestamp': datetime.datetime(2022, 3, 30, 23, 26, 33, 499000,
tzinfo=tzlocal())
 },
 ...
]
}

```

## Elenca i cluster

Recupera un elenco di cluster esistenti.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)



- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
GET /v3/clusters
{
 "clusterStatus": "string",
 "nextToken": "string",
 "region": "string"
}
```

## Corpo di richiesta

### ClusterStatus

Filtra per stato del cluster. L'impostazione predefinita è tutti i cluster.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_FAILED

Obbligatorio: no

### nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

### Regione

Regione AWS I cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "clusters": [
 {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
 }
]
}
```

## Corpo di risposta

### cluster

#### cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

#### cloudformationStackStatus

Lo stato dello CloudFormation stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS

| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

Nome del cluster

Il nome del cluster.

Tipo: stringa

Stato del cluster

Lo stato del cluster.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

pianificatore

metadata

I metadati dello scheduler.

nome

Il nome dello scheduler.

Tipo: stringa

version

La versione dello scheduler.

Tipo: stringa

tipo

Il tipo di scheduler.

Tipo: stringa

Regione

Regione AWS In cui viene creato il cluster.

Tipo: stringa

## version

La AWS ParallelCluster versione utilizzata per creare il cluster.

Tipo: stringa

## nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ list_clusters()
```

#### 200 Risposta

```
{
 'clusters':
 [
 {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cloudformation_stack_status': 'CREATE_COMPLETE',
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'CREATE_COMPLETE',
 'region': 'us-east-1',
 'version': '3.2.1'
 },
 ...
]
}
```

## listClusterLogStream

Recupera l'elenco dei flussi di log associati a un cluster.

## Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
GET /v3/clusters/{clusterName}/logstreams
{
 "filters": ["string"],
 "nextToken": "string",
 "region": "string"
}
```

## Corpo di richiesta

### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

### filtri

Filtra i flussi di log.

I filtri accettati sono:

- `private-dns-name`: La forma abbreviata del nome DNS privato dell'istanza (ad esempio `ip-10-0-0-101`).
- `node-type`: Valore valido: `HeadNode`

Tipo: matrice di stringhe unica

Formato: `Name=a,Values=1 Name=b,Values=2,3`

Campo obbligatorio: no

nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "logStreams": [
 {
 "logStreamName": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "firstEventTimestamp": "2019-08-24T14:15:22Z",
 "lastEventTimestamp": "2019-08-24T14:15:22Z",
 "lastIngestionTime": "2019-08-24T14:15:22Z",
 "uploadSequenceToken": "string",
 "logStreamArn": "string"
 }
]
}
```

## Corpo di risposta

### LogStreams

Un elenco di flussi di log.

creationTime

L'ora in cui è stato creato lo stream.

Tipo: datetime

firstEventTimestamp

L'ora del primo evento dello stream.

Tipo: datetime

lastEventTimestamp

L'ora dell'ultimo evento dello stream. Il lastEventTime valore si aggiorna in base alla coerenza finale. In genere si aggiorna in meno di un'ora dall'inserimento, ma in rare situazioni potrebbe richiedere più tempo.

Tipo: datetime

lastIngestionTime

L'ora dell'ultima ingestione.

Tipo: datetime

logStreamArn

L'Amazon Resource Name (ARN) del flusso di log.

Tipo: stringa

logStreamName

Nome del flusso di log.

Tipo: stringa

uploadSequenceToken

Il token della sequenza.

Tipo: stringa

nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure null se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ list_cluster_log_streams(cluster_name_3x)
```

#### 200 Risposta

```
{
 'log_streams': [
 {
 'creation_time': datetime.datetime(2022, 3, 30, 14, 7, 34, 354000,
tzinfo=tzlocal()),
 'first_event_timestamp': datetime.datetime(2022, 3, 30, 14, 6, 41, 444000,
tzinfo=tzlocal()),
 'last_event_timestamp': datetime.datetime(2022, 3, 30, 14, 25, 55, 462000,
tzinfo=tzlocal()),
 'last_ingestion_time': datetime.datetime(2022, 3, 30, 14, 49, 50, 62000,
tzinfo=tzlocal()),
 'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
parallelcluster/cluster_name_3x:log-stream:ip-192-0-2-26.i-abcdef01234567890.cfn-
init',
 'log_stream_name': 'ip-192-0-2-26.i-abcdef01234567890.cfn-init',
 ...
 'upload_sequence_token': '####'
 },
 ...
]
}
```

## listImageLogStream

Recupera l'elenco dei flussi di log associati a un'immagine.

#### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)



- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
GET /v3/images/custom/{imageId}/logstreams
{
 "nextToken": "string",
 "region": "string"
}
```

## Corpo di richiesta

### imageId

L'ID dell'immagine.

Tipo: stringa

Campo obbligatorio: sì

### nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

### Regione

Regione AWS Quello in cui si trova l'immagine.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
```

```
"logStreams": [
 {
 "logStreamName": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "firstEventTimestamp": "2019-08-24T14:15:22Z",
 "lastEventTimestamp": "2019-08-24T14:15:22Z",
 "lastIngestionTime": "2019-08-24T14:15:22Z",
 "uploadSequenceToken": "string",
 "logStreamArn": "string"
 }
]
}
```

## Corpo di risposta

### Registra i flussi

Un elenco di flussi di log.

`creationTime`

L'ora in cui è stato creato lo stream.

Tipo: `datetime`

`firstEventTimestamp`

L'ora del primo evento nello stream.

Tipo: `datetime`

`lastEventTimestamp`

L'ora dell'ultimo evento dello stream. Il `lastEventTime` valore si aggiorna in base alla coerenza finale. In genere si aggiorna in meno di un'ora dall'inserimento, ma in rare situazioni potrebbe richiedere più tempo.

Tipo: `datetime`

`lastIngestionTime`

L'ora dell'ultima ingestione.

Tipo: `datetime`

## logStreamArn

L'Amazon Resource Name (ARN) del flusso di log.

Tipo: stringa

## logStreamName

Il nome del flusso di log.

Tipo: stringa

## uploadSequenceToken

Il token di sequenza.

Tipo: stringa

## next\_token

Token che è possibile usare per recuperare il set di risultati successivo oppure `null` se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ list_image_log_streams(custom-image-id)
```

#### 200 Risposta

```
{
 'log_streams': [
 {
 'creation_time': datetime.datetime(2022, 3, 29, 20, 29, 24, 875000,
tzinfo=tzlocal()),
 'first_event_timestamp': datetime.datetime(2022, 3, 29, 20, 29, 24, 775000,
tzinfo=tzlocal()),
 'last_event_timestamp': datetime.datetime(2022, 3, 29, 20, 38, 23, 944000,
tzinfo=tzlocal()),
```

```
 'last_ingestion_time': datetime.datetime(2022, 3, 29, 20, 51, 56, 26000,
 tzinfo=tzlocal()),
 'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image:log-stream:3.2.1/1',
 'log_stream_name': '3.2.1/1',
 'upload_sequence_token': '####'
},
...
]
```

## Elenca immagini

Recupera l'elenco delle immagini personalizzate esistenti.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
GET /images/custom
{
 "imageStatus": "string",
 "nextToken": "string",
 "region": "string"
}
```

### Corpo di richiesta

#### Stato dell'immagine

Filtra le immagini in base allo stato fornito.

Tipo: stringa

Valori validi: AVAILABLE | PENDING | FAILED

Campo obbligatorio: sì

nextToken

Token per il set di risultati successivo.

Tipo: stringa

Campo obbligatorio: no

Regione

Le immagini in Regione AWS cui sono inserite.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "nextToken": "string",
 "images": [
 {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS"
 }
]
}
```

## Corpo di risposta

images

Un elenco di immagini.

## cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

## cloudformationStackStatus

Lo stato dello CloudFormation stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmiInfo

### ami\_id

L'ID Amazon EC2 AMI.

Tipo: stringa

## imageBuildStatus

Lo stato di creazione dell'immagine.

Valori validi: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

Tipo: stringa

## imageId

L'ID dell'immagine.

Tipo: stringa

## Regione

L' Regione AWS elemento in cui viene creata l'immagine.

Tipo: stringa

version

La AWS ParallelCluster versione utilizzata per creare l'immagine.

Tipo: stringa

nextToken

Token che è possibile usare per recuperare il set di risultati successivo oppure null se non ci sono risultati aggiuntivi.

Tipo: stringa

## Esempio

Python

Richiesta

```
$ list_images("AVAILABLE")
```

200 Risposta

```
{
 'images': [
 {
 'ec2_ami_info': {
 'ami_id': 'ami-abcdef01234567890'
 },
 'image_build_status': 'BUILD_COMPLETE',
 'image_id': 'custom-image',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
]
}
```

## listOfficialImages

Recupera l'elenco delle immagini AWS ParallelCluster ufficiali.

## Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

## Sintassi della richiesta

```
GET /v3/images/official
{
 "architecture": "string",
 "os": "string",
 "region": "string"
}
```

## Corpo di richiesta

### architecture

Filtra per architettura. L'impostazione predefinita è nessun filtro.

Tipo: stringa

Valori validi: x86\_64 | arm64

Campo obbligatorio: no

### so

Filtra per distribuzione del sistema operativo. L'impostazione predefinita è nessun filtro.

Tipo: stringa

Valori validi: alinux2 | alinux2023 | ubuntu2404 | ubuntu2204 | rhel8 | rhel9

Campo obbligatorio: no

## Regione

La sezione Regione AWS in cui sono elencate le immagini ufficiali.



Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "images": [
 {
 "architecture": "string",
 "amiId": "string",
 "name": "string",
 "os": "string",
 "version": "string"
 }
]
}
```

## Corpo di risposta

images

AmiID

L'ID dell'AMI.

Tipo: stringa

architecture

L'architettura AMI.

Tipo: stringa

nome

Il nome dell'AMI.

Tipo: stringa

so

Il sistema operativo AMI.

Tipo: stringa

## version

La AWS ParallelCluster versione.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ list_official_images()
```

#### 200 Risposta

```
{
 'images': [
 {
 'ami_id': 'ami-015cfef4e0d6306b2',
 'architecture': 'x86_64',
 'name': 'aws-parallelcluster-3.2.1-ubuntu-2204-lts-hvm-x86_64-202202261505 '
 '2022-02-26T15-08-34.759Z',
 'os': 'ubuntu2204',
 'version': '3.2.1'
 },
 ...
]
}
```

## Aggiorna Cluster

Aggiorna il cluster.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)

- [Esempio](#)

## Sintassi della richiesta

```
PUT /v3/clusters/{clusterName}
{
 "clusterConfiguration": "string",
 "dryrun": boolean,
 "forceUpdate": boolean,
 "region": "string",
 "suppressValidators": "string",
 "validationFailureLevel": "string"
}
```

## Corpo di richiesta

### Configurazione del cluster

La configurazione del cluster come documento YAML.

Campo obbligatorio: sì

### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

### funzionamento a secco

Se impostato su `true`, esegue solo la convalida della richiesta senza creare alcuna risorsa.

Utilizzate questo parametro per convalidare la configurazione del cluster e aggiornare i requisiti. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

### ForceUpdate

Se impostato su `true`, ignora gli errori di convalida dell'aggiornamento e forza l'aggiornamento. Il valore predefinito è `false`.

Tipo: booleano

Campo obbligatorio: no

## Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sopprimere i validatori

Identifica uno o più validatori di configurazione da sopprimere.

Tipo: stringa

Formato: (ALL | type: [A-Za-z0-9]+)

Campo obbligatorio: no

Esempi di valori validi:,, currentValue requestedValue message

## validationFailureLevel

Il livello di convalida minimo per impedire l'aggiornamento.

Tipo: stringa

Valori validi: INFO | WARNING | ERROR

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "clusterStatus": "UPDATE_IN_PROGRESS",
 "scheduler": {
```

```
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 },
 "validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
],
 "changeSet": [
 {
 "parameter": "string",
 "currentValue": "string",
 "requestedValue": "string"
 }
]
}
```

## Corpo di risposta

### changeSet

Il set di modifiche per l'aggiornamento del cluster.

Valore corrente

Il valore corrente del parametro da aggiornare.

Tipo: stringa

parametro

Il parametro da aggiornare.

Tipo: stringa

Valore richiesto

Il valore richiesto per il parametro da aggiornare.

Tipo: stringa

cluster

cloudformationStackArn

L'Amazon Resource Name (ARN) dello stack principale CloudFormation .

Tipo: stringa

cloudformationStackStatus

Lo stato dello CloudFormation stack.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

Nome del cluster

Il nome del cluster.

Tipo: stringa

ClusterStatus

Lo stato del cluster.

Tipo: stringa

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

Regione

Lo Regione AWS in cui viene creato il cluster.

Tipo: stringa

## pianificatore

### metadata

I metadati dello scheduler.

#### nome

Il nome dello scheduler.

Tipo: stringa

#### version

La versione dello scheduler.

Tipo: stringa

### tipo

Il tipo di scheduler.

Tipo: stringa

### version

AWS ParallelCluster versione utilizzata per creare il cluster.

Tipo: stringa

## Messaggi di convalida

Un elenco di messaggi con un livello di convalida inferiore a `validationFailureLevel`.  
L'elenco dei messaggi viene raccolto durante la convalida della configurazione.

### id

L'ID del validatore.

Tipo: stringa

### level

Il livello di convalida.

Tipo: stringa

Valori validi: INFO | WARNING | ERROR

## message

Il messaggio di convalida.

Tipo: stringa

## tipo

Il tipo di validatore.

Tipo: stringa

## Esempio

### Python

#### Richiesta

```
$ update_cluster(cluster_name_3x, path/config-file.yaml)
```

#### 200 Risposta

```
{
 'change_set': [
 {
 'current_value': '10',
 'parameter':
'Scheduling.SlurmQueues[queue1].ComputeResources[t2micro].MaxCount',
 'requested_value': '15'
 }
],
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/test-api-cluster/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
 'cloudformation_stack_status': 'UPDATE_IN_PROGRESS',
 'cluster_name': 'cluster-3x',
 'cluster_status': 'UPDATE_IN_PROGRESS',
 'region': 'us-east-1',
 'scheduler': {
 'type': 'slurm'
 },
 'version': '3.2.1'
 }
}
```



```
}
```

## updateComputeFleet

Aggiorna lo stato della flotta di elaborazione del cluster.

### Argomenti

- [Sintassi della richiesta](#)
- [Corpo di richiesta](#)
- [Sintassi della risposta](#)
- [Corpo di risposta](#)
- [Esempio](#)

### Sintassi della richiesta

```
PATCH /v3/clusters/{clusterName}/computefleet
{
 "status": "string",
 "region": "string"
}
```

### Corpo di richiesta

#### ClusterName

Il nome del cluster.

Tipo: stringa

Campo obbligatorio: sì

#### stato

Lo stato della flotta di elaborazione.

Tipo: stringa

Valori validi: START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

Campo obbligatorio: sì

## Regione

Regione AWS Quello in cui si trova il cluster.

Tipo: stringa

Campo obbligatorio: no

## Sintassi della risposta

```
{
 "status": "START_REQUESTED",
 "lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

## Corpo di risposta

### stato

Lo stato della flotta di elaborazione.

Tipo: stringa

Valori validi: START\_REQUESTED | STARTING | RUNNING | PROTECTED |  
STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdatedOra

Il timestamp che rappresenta l'ora dell'ultimo aggiornamento dello stato.

Tipo: datetime

## Esempio

### Python

#### Richiesta

```
$ update_compute_fleet(cluster_name_3x, "START_REQUESTED")
```

## 200 Risposta

```
{
 'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
 tzinfo=tzlocal()),
 'status': 'START_REQUESTED'
}
```

## AWS ParallelCluster API della libreria Python

A partire dalla AWS ParallelCluster versione 3.5.0, è possibile accedere AWS ParallelCluster con la libreria AWS ParallelCluster Python. È possibile accedere alla AWS ParallelCluster libreria nel proprio `pcluster` ambiente o dall'interno di un AWS Lambda runtime. Scopri come accedere all' AWS ParallelCluster API utilizzando la libreria AWS ParallelCluster Python. La libreria AWS ParallelCluster Python offre le stesse funzionalità offerte dall' AWS ParallelCluster API.

Le operazioni e i parametri della libreria AWS ParallelCluster Python rispecchiano quelli dei parametri API quando vengono convertiti in caratteri `snake_case` senza lettere maiuscole.

### Argomenti

- [AWS ParallelCluster Autorizzazione della libreria Python](#)
- [Installa la AWS ParallelCluster libreria Python](#)
- [Operazioni dell'API del cluster](#)
- [Elabora le operazioni API della flotta](#)
- [Operazioni di cluster e stack log](#)
- [Operazioni dell'API di immagine](#)
- [Operazioni relative ai log delle immagini e degli stack](#)
- [Esempio](#)
- [AWS Lambda per la libreria AWS ParallelCluster Python](#)

## AWS ParallelCluster Autorizzazione della libreria Python

Specificate le credenziali utilizzando uno dei metodi standard validi per boto3. [Per ulteriori informazioni, consultate la documentazione di boto3.](#)

## Installa la AWS ParallelCluster libreria Python

1. Installa la versione `pcluster` CLI 3.5.0 o successiva seguendo le istruzioni fornite in.  
[Configurazione AWS ParallelCluster](#)
2. Importa il `pcluster` modulo e inizia a utilizzare la libreria, come mostrato nell'esempio seguente:

```
import pcluster.lib as pc
pc.create_cluster(cluster_name="mycluster", cluster_configuration="config.yaml")
```

## Operazioni dell'API del cluster

### Argomenti

- [list\\_clusters](#)
- [create\\_cluster](#)
- [delete\\_cluster](#)
- [describe\\_cluster](#)
- [update\\_cluster](#)

## list\_clusters

```
list_clusters(region, next_token, cluster_status)
```

Recupera l'elenco dei cluster esistenti.

Parametri:

### **region**

Elenca i cluster distribuiti in un determinato ambiente. Regione AWS

### **next\_token**

Token per il set di risultati successivo.

### **cluster\_status**

Filtri in base allo stato del cluster. L'impostazione predefinita prevede l'elenco di tutti i cluster.

Valori validi: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE |  
UPDATE\_FAILED

## create\_cluster

```
create_cluster(cluster_name, cluster_configuration, region, suppress_validators,
validation_failure_level, dry_run, rollback_on_failure, wait)
```

Crea un cluster in una determinata regione.

Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

**cluster\_configuration**(richiesto)

La configurazione del cluster come tipo di dati Python.

**region**

Il cluster Regione AWS.

**suppress\_validators**

Identifica uno o più validatori di configurazione del cluster da sopprimere.

Formato: (ALL | type:[A-Za-z0-9]+)

**validation\_failure\_level**

Il livello di convalida minimo che causa l'esito negativo della creazione del cluster. Il valore predefinito è ERROR.

Valori validi: INFO | WARNING | ERROR.

**dry\_run**

Esegue la convalida della richiesta senza creare risorse. È possibile utilizzarlo per convalidare la configurazione del cluster. Il valore predefinito è False.

## **rollback\_on\_failure**

Se impostato su `True`, avvia AWS ParallelCluster automaticamente un rollback dello stack del cluster in caso di errori. Il valore predefinito è `True`.

## **wait**

Se impostato su `True`, AWS ParallelCluster attende il completamento dell'operazione. Il valore predefinito è `False`.

## **delete\_cluster**

```
delete_cluster(cluster_name, region, wait)
```

Elimina un cluster in una determinata regione.

Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

**region**

Il cluster Regione AWS.

**wait**

Se impostato su `True`, attende il completamento dell'operazione. Il valore predefinito è `False`.

## **describe\_cluster**

```
describe_cluster(cluster_name, region)
```

Ottieni informazioni dettagliate su un cluster esistente.

Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

## **region**

Il cluster Regione AWS.

## **update\_cluster**

```
update_cluster(cluster_name, cluster_configuration, suppress_validators,
validation_failure_level, region, force_update, dry_run, wait)
```

Aggiorna un cluster in una determinata regione.

Parametri:

### **cluster\_name**(richiesto)

Il nome del cluster.

### **cluster\_configuration**(richiesto)

La configurazione del cluster come tipo di dati Python.

### **suppress\_validators**

Identifica uno o più validatori di configurazione del cluster da sopprimere.

Formato: (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

Il livello di convalida minimo che causa il fallimento dell'aggiornamento del cluster. Il valore predefinito è ERROR.

Valori validi: INFO | WARNING | ERROR

## **region**

Il cluster Regione AWS.

## **dry\_run**

Esegue la convalida della richiesta senza creare o aggiornare alcuna risorsa. È possibile utilizzarlo per convalidare la configurazione del cluster. Il valore predefinito è False.

## **force\_update**

Se impostato su `True`, forza l'aggiornamento ignorando gli errori di convalida dell'aggiornamento. Il valore predefinito è `False`.

## **wait**

Se impostato su `True`, attende il completamento dell'operazione. Il valore predefinito è `False`.

# Elabora le operazioni API della flotta

## Argomenti

- [describe\\_compute\\_fleet](#)
- [update\\_compute\\_fleet](#)
- [delete\\_cluster\\_instances](#)
- [describe\\_cluster\\_instances](#)

## **describe\_compute\_fleet**

```
describe_compute_fleet(cluster_name, region)
```

Descrivi lo stato di una flotta di elaborazione del cluster per un determinato cluster.

## Parametri:

### **cluster\_name**(richiesto)

Il nome del cluster.

### **region**

Descrive lo stato della flotta di elaborazione per un cluster distribuito in un determinato ambiente. Regione AWS

## **update\_compute\_fleet**

```
update_compute_fleet(cluster_name, status, region)
```

Aggiorna lo stato della flotta di elaborazione del cluster.



Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

**status**(richiesto)

Lo stato a cui eseguire l'aggiornamento.

Valori validi: START\_REQUESTED| STOP\_REQUESTED| ENABLED| DISABLED

**region**

Il cluster Regione AWS.

## delete\_cluster\_instances

```
delete_cluster_instances(cluster_name, region, force)
```

Avvia la chiusura forzata di tutti i nodi di calcolo del cluster. Questa azione non supporta i cluster AWS Batch

Parametri:

**cluster\_name**(obbligatorio)

Il nome del cluster.

**region**

Il cluster Regione AWS.

**force**

Se impostato su `True`, forza l'eliminazione quando il cluster con il dato dato `cluster_name` non viene trovato. Il valore predefinito è `False`.

## describe\_cluster\_instances

```
describe_cluster_instances(cluster_name, region, next_token, node_type, queue_name)
```

Descrivi le istanze di un cluster.

Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

**region**

Il cluster Regione AWS.

**next\_token**

Token per il set di risultati successivo.

**node\_type**

Filtra le istanze per node\_type.

Valori validi: HeadNode | ComputeNode

**queue\_name**

Filtra le istanze in base al nome della coda.

## Operazioni di cluster e stack log

Argomenti

- [list\\_cluster\\_log\\_streams](#)
- [get\\_cluster\\_log\\_events](#)
- [get\\_cluster\\_stack\\_events](#)

### list\_cluster\_log\_streams

```
list_cluster_log_streams(cluster_name, region, filters, next_token)
```

Elenca i flussi di log per un determinato cluster.

Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

## **region**

Il cluster Regione AWS.

## **filters**

Filtra i flussi di log del cluster.

Formato: 'Name=a,Values=1 Name=b,Values=2,3'

Filtri accettati:

code-dns-name

La forma abbreviata del nome DNS privato dell'istanza, ip-10-0-0-101 ad esempio.

tipo di nodo

Il tipo di nodo.

Valori validi: HeadNode

## **next\_token**

Token per il set di risultati successivo.

## **get\_cluster\_log\_events**

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,
start_from_head, limit, start_time, end_time)
```

Ottieni gli eventi di registro per un determinato cluster e flusso di log.

Parametri:

**cluster\_name**(richiesto)

Il nome del cluster.

**log\_stream\_name**(richiesto)

Il nome del flusso di log.

## **region**

Il cluster Regione AWS.

## **next\_token**

Token per il set di risultati successivo.

## **start\_from\_head**

Se impostato su `True`, AWS ParallelCluster restituisce per primi i primi eventi di registro. Se impostato su `False`, restituisce per primi gli ultimi eventi di registro. Il valore predefinito è `False`.

## **limit**

Il numero massimo di eventi di registro restituiti. Se non si specifica un valore, il numero massimo è il numero di log che possono contenere una dimensione di risposta di 1 MB, fino a 10.000 eventi di registro.

## **start\_time**

L'inizio dell'intervallo di tempo per gli eventi di registro, espresso in formato ISO 8601; ad esempio, `'2021-01-01T20:00:00Z'`. Sono inclusi gli eventi con un timestamp uguale o successivo a questo orario.

## **end\_time**

La fine dell'intervallo di tempo per gli eventi di registro, espressa nel formato ISO 8601; ad esempio, `'2021-01-01T20:00:00Z'`. Gli eventi con un timestamp uguale o successivo a questo orario non sono inclusi.

# **get\_cluster\_stack\_events**

```
get_cluster_stack_events(cluster_name, region, next_token)
```

Ottieni gli eventi dello stack per un determinato cluster.

Parametri:

## **cluster\_name**(richiesto)

Il nome del cluster.

## **region**

Il cluster Regione AWS.

## next\_token

Token per il set di risultati successivo.

## Operazioni dell'API di immagine

### Argomenti

- [list\\_images](#)
- [build\\_image](#)
- [delete\\_image](#)
- [describe\\_image](#)

## list\_images

```
list_images(image_status, region, next_token)
```

Recupera l'elenco delle immagini esistenti.

Parametri:

### **image\_status**(richiesto)

Filtri per stato dell'immagine.

Valori validi: AVAILABLE | PENDING | FAILED

### **region**

Elenca le immagini incorporate in un determinato elemento Regione AWS.

## next\_token

Token per il set di risultati successivo.

## build\_image

```
build_image(image_configuration, image_id, suppress_validators,
validation_failure_level, dry_run, rollback_on_failure, region)
```

Crea un' AWS ParallelCluster immagine personalizzata in una determinata regione.

Parametri:

### **image\_configuration**(richiesto)

La configurazione dell'immagine come dati Python.

### **image\_id**(richiesto)

L'ID dell'immagine.

### **suppress\_validators**

Identifica uno o più validatori di configurazione delle immagini da sopprimere.

Formato: (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

Il livello di convalida minimo che causa il fallimento della creazione dell'immagine. Il valore predefinito è ERROR.

Valori validi: INFO | WARNING | ERROR

### **dry\_run**

Se impostato su True, AWS ParallelCluster esegue la convalida della richiesta senza creare risorse. È possibile utilizzarlo per convalidare la configurazione dell'immagine. Il valore predefinito è False.

### **rollback\_on\_failure**

Se impostato su True, avvia AWS ParallelCluster automaticamente un rollback dello stack di immagini in caso di errori. Il valore predefinito è False.

### **region**

L'immagine. Regione AWS

## **delete\_image**

```
delete_image(image_id, region, force)
```

Elimina un'immagine in una determinata regione.

Parametri:

**image\_id**(richiesto)

L'ID dell'immagine.

**region**

L'immagine Regione AWS.

**force**

Se impostato su `True`, AWS ParallelCluster forza l'eliminazione se le istanze utilizzano l'AMI o se l'AMI è condiviso. Il valore predefinito è `False`.

## describe\_image

```
describe_image(image_id, region)
```

Ottieni informazioni dettagliate su un'immagine esistente.

Parametri:

**image\_id**(richiesto)

L'ID dell'immagine.

**region**

L'immagine Regione AWS.

## Operazioni relative ai log delle immagini e degli stack

Argomenti

- [list\\_image\\_log\\_streams](#)
- [get\\_image\\_log\\_events](#)
- [get\\_image\\_stack\\_events](#)

- [list\\_official\\_images](#)

## list\_image\_log\_streams

```
list_image_log_streams(image_id, region, next_token)
```

Elenca i flussi di registro per un'immagine.

Parametri:

**image\_id**(richiesto)

L'ID dell'immagine.

**region**

L'immagine Regione AWS.

**next\_token**

Token per il set di risultati successivo.

## get\_image\_log\_events

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head, limit, start_time, end_time)
```

Ottieni gli eventi di registro per una determinata immagine e flusso di log.

Parametri:

**image\_id**(richiesto)

L'ID dell'immagine.

**log\_stream\_name**(richiesto)

Il nome del flusso di log.

**region**

L'immagine Regione AWS.



## **next\_token**

Token per il set di risultati successivo.

## **start\_from\_head**

Se impostato su `True`, AWS ParallelCluster restituisce per primi i primi eventi di registro. Se impostato su `False`, restituisce per primi gli ultimi eventi di registro. Il valore predefinito è `False`.

## **limit**

Il numero massimo di eventi di registro restituiti. Se non si specifica un valore, il numero massimo è il numero di log che possono contenere una dimensione di risposta di 1 MB, fino a 10.000 eventi di registro.

## **start\_time**

L'inizio dell'intervallo di tempo per gli eventi di registro, espresso in formato ISO 8601; ad esempio, '2021-01-01T20:00:00Z'. Sono inclusi gli eventi con un timestamp uguale o successivo a questo orario.

## **end\_time**

La fine dell'intervallo di tempo per gli eventi di registro, espressa nel formato ISO 8601; ad esempio, '2021-01-01T20:00:00Z'. Gli eventi con un timestamp uguale o successivo a questo orario non sono inclusi.

## **get\_image\_stack\_events**

```
get_image_stack_events(image_id, region, next_token)
```

Ottieni gli eventi dello stack per una determinata immagine.

Parametri:

### **image\_id**(richiesto)

L'ID dell'immagine.

### **region**

L'immagine Regione AWS.

## **next\_token**

Token per il set di risultati successivo.

## **list\_official\_images**

```
list_official_images(region,os, architecture)
```

Recupera l'elenco delle immagini ufficiali AWS ParallelCluster .

Parametri:

### **region**

L'immagine Regione AWS.

### **os**

Filtri per distribuzione del sistema operativo. L'impostazione predefinita è nessun filtro.

### **architecture**

Filtri per architettura. L'impostazione predefinita è nessun filtro.

## Esempio

Argomenti

- [Creazione di un cluster](#)

### Creazione di un cluster

Quando si esegue lo script di esempio seguente, con gli input forniti memorizzati nel proprio ambiente, si crea un cluster. La configurazione del cluster viene creata come tipo di dati Python in base alla documentazione di [configurazione del cluster](#).

```
import os
import pprint
import pcluster.lib as pc
pp = pprint.PrettyPrinter()

HEAD_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
```

```

COMPUTE_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
KEY_NAME = os.environ["KEY_NAME"]
CONFIG = {'Image': {'Os': 'alinux2'},
 'HeadNode': {'InstanceType': 't2.large',
 'Networking': {'SubnetId': HEAD_NODE_SUBNET},
 'Ssh': {'KeyName': KEY_NAME}},

 'Scheduling': {'Scheduler': 'slurm',
 'SlurmQueues':
 [{ 'Name': 'queue0',
 'ComputeResources':
 [{ 'Name': 'queue0-i0', 'InstanceType': 't2.micro',
 'MinCount': 0, 'MaxCount': 10}],
 'Networking': {'SubnetIds': [COMPUTE_NODE_SUBNET]}]}}}

pp.pprint(pc.create_cluster(cluster_name="mycluster", cluster_configuration=CONFIG))

```

Output:

```

{'cluster': {'cloudformationStackArn': 'arn:aws:cloudformation:us-
east-2:123456789012:stack/mycluster/000000000-aaaa-1111-999-0000000000000',
 'cloudformationStackStatus': 'CREATE_IN_PROGRESS',
 'clusterName': 'mycluster',
 'clusterStatus': 'CREATE_IN_PROGRESS',
 'region': 'us-east-2',
 'scheduler': {'type': 'slurm'},
 'version': '3.15.0'}}

```

## AWS Lambda per la libreria AWS ParallelCluster Python

Puoi implementare un layer Lambda e un runtime per accedere alla libreria AWS ParallelCluster Python. Ospitiamo file AWS ParallelCluster zip che puoi utilizzare inserendo il link al file zip come descritto nei passaggi seguenti. Lambda utilizza i file zip per preparare l'ambiente di runtime per supportare l'accesso alla libreria Python. La libreria AWS ParallelCluster Python viene aggiunta con la AWS ParallelCluster versione 3.5.0. È possibile utilizzare la libreria solo per le versioni 3.5.0 e successive.

L'URL del file zip ospitato è nel formato: `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.15.0/layers/aws-parallelcluster/lambda-layer.zip`. (Sostituisci **3.15.0** con la AWS ParallelCluster versione che desideri utilizzare nel passaggio successivo.)

## Inizia ad accedere alla libreria AWS ParallelCluster Python con AWS Lambda

### Creare un layer Lambda

1. Accedi a Console di gestione AWS e vai alla AWS Lambda console.
2. Nel riquadro di navigazione, seleziona Livelli, quindi Crea livello.
3. Inserisci un nome per il tuo layer e seleziona Carica un file da Amazon S3.
4. Inserisci l'URL del file zip: `s3://aws-region-id3.15.0-aws-parallelcluster/parallelcluster//layer.zip`. layers/aws-parallelcluster/lambda
5. Per le architetture compatibili, scegli l'architettura x86\_64.
6. Per i runtime compatibili, scegli il runtime Python 3.12.
7. Scegli Create (Crea).

### Usa il tuo layer Lambda

1. Nel pannello di navigazione della console Lambda, seleziona Funzioni, quindi Crea funzione.
2. Immetti un nome per la funzione.
3. Per Runtime, scegli il runtime Python 3.12.
4. Per Architettura, scegli l'architettura x86\_64.
5. Scegli Crea funzione.
6. Dopo aver creato la funzione, scegliete Livelli e selezionate Aggiungi un livello.
7. Seleziona Livelli personalizzati e scegli il livello creato nei passaggi precedenti.
8. Scegliete la versione del livello.
9. Scegliere Aggiungi.
10. La tua Lambda necessita delle autorizzazioni per gestire i cluster creati con. AWS ParallelCluster Crea un ruolo Lambda con le autorizzazioni elencate in. [Politica AWS ParallelCluster pcluster utente di base](#)

Ora puoi accedere AWS ParallelCluster dalla libreria Python come descritto in. [AWS ParallelCluster API della libreria Python](#)

# Tutorial su come usare AWS ParallelCluster

I seguenti tutorial mostrano come iniziare a usare la AWS ParallelCluster versione 3 e forniscono indicazioni sulle migliori pratiche per alcune attività comuni.

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

## Argomenti

- [Esecuzione del tuo primo lavoro su AWS ParallelCluster](#)
- [Creazione di un' AWS ParallelCluster AMI personalizzata](#)
- [Integrazione di Active Directory](#)
- [Configurazione della crittografia dello storage condiviso con una chiave AWS KMS](#)
- [Esecuzione di processi in un cluster in modalità coda multipla](#)
- [Utilizzo dell' AWS ParallelCluster API](#)
- [Creazione di un cluster con Slurm contabilità](#)
- [Creazione di un cluster con un cluster esterno Slurmdbd contabilità](#)
- [Ripristino a una versione precedente del documento AWS Systems Manager](#)
- [Creazione di un cluster con CloudFormation](#)
- [Implementa l' ParallelCluster API con Terraform](#)
- [Creare un cluster con Terraform](#)
- [Creazione di un'AMI personalizzata con Terraform](#)
- [AWS ParallelCluster Integrazione dell'interfaccia utente con Identity Center](#)
- [Esecuzione di lavori containerizzati con Pyxis](#)
- [Creazione di un cluster con un Lustre abilitato per EFA FSx](#)
- [Supporta Nvidia-IMEX con istanza p6e-gb200](#)
- [Personalizza le interfacce di rete dei nodi di calcolo con sostituzioni dei modelli di avvio](#)

## Esecuzione del tuo primo lavoro su AWS ParallelCluster

Questo tutorial ti spiega come eseguire il tuo primo lavoro in Hello World su AWS ParallelCluster

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

## Prerequisiti

- AWS ParallelCluster [è](#) installato.
- AWS CLI [è installato e configurato](#).
- Hai una coppia di [chiavi Amazon EC2](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

## Verifica dell'installazione

Innanzitutto, verifichiamo che AWS ParallelCluster sia installato e configurato correttamente, inclusa la dipendenza Node.js.

```
$ node --version
v16.8.0
$ pcluster version
{
 "version": "3.15.0"
}
```

Ciò restituisce la versione in esecuzione di. AWS ParallelCluster

## Creazione del primo cluster

Verrà ora creato il tuo primo cluster. Poiché il carico di lavoro per questo tutorial non ha elevati requisiti di prestazioni, possiamo utilizzare la dimensione dell'istanza predefinita di `t2.micro`. (Per carichi di lavoro di produzione, scegli la dimensione dell'istanza più adatta alle tue esigenze.) Chiamiamo il tuo cluster `hello-world`.

```
$ pcluster create-cluster \
 --cluster-name hello-world \
 --cluster-configuration hello-world.yaml
```

 Note

La modalità Regione AWS di utilizzo deve essere specificata per la maggior parte dei `pcluster` comandi. Se non è specificato nella variabile di `AWS_DEFAULT_REGION` ambiente o nell'impostazione nella `[default]` sezione del `~/.aws/config` file, il `--region` parametro deve essere fornito nella riga di `pcluster` comando.

Se l'output mostra un messaggio sulla configurazione, devi eseguire quanto segue per configurare AWS ParallelCluster:

```
$ pcluster configure --config hello-world.yaml
```

Se il [pcluster create-cluster](#) comando ha esito positivo, viene visualizzato un output simile al seguente:

```
{
 "cluster": {
 "clusterName": "hello-world",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:xxx:stack/xxx",
 "region": "...",
 "version": "...",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

È possibile monitorare la creazione del cluster utilizzando:

```
$ pcluster describe-cluster --cluster-name hello-world
```

Il `clusterStatus` report `"CREATE_IN_PROGRESS"` durante la creazione del cluster. Le `clusterStatus` transizioni a `"CREATE_COMPLETE"` quando il cluster viene creato correttamente. L'output ci fornisce anche la `privateIpAddress` del `publicIpAddress` nostro nodo principale.

## Accedere al nodo principale

Usa il tuo file pem OpenSSH per accedere al tuo nodo principale.

```
$ pcluster ssh --cluster-name hello-world -i /path/to/keyfile.pem
```

Dopo aver effettuato l'accesso, esegui il comando `sinfo` per verificare che i nodi di calcolo siano impostati e configurati.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 10 idle~ queue1-dy-queue1t2micro-[1-10]
```

L'output mostra che abbiamo una coda nel nostro cluster, con un massimo di dieci nodi.

## Esecuzione del primo lavoro con Slurm

Creiamo un processo che rimane in sospeso per alcuni secondi e che quindi fornisce in uscita il suo nome host. Crea un file denominato `hellojob.sh`, con i seguenti contenuti:

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

Quindi, invia il processo utilizzando `sbatch` e verifica che sia in esecuzione.

```
$ sbatch hellojob.sh
Submitted batch job 2
```

Puoi ora visualizzare la coda e verificare lo stato del processo. Il provisioning di una nuova istanza Amazon EC2 viene avviato in background. Puoi monitorare lo stato delle istanze del cluster con il comando `sinfo`

```
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 queue1 hellojob ec2-user CF 3:30 1 queue1-dy-
queue1t2micro-1
```

L'output mostra che il lavoro è stato inviato a `queue1`. Attendi 30 secondi per il completamento del processo, quindi esegui nuovamente `squeue`.

```
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 queue1 hellojob ec2-user CF 3:30 1 queue1-dy-
queue1t2micro-1
```



Ora che non ci sono processi in coda, puoi verificare l'output nella directory corrente.

```
$ ls -l
total 8
-rw-rw-r-- 1 ec2-user ec2-user 57 Sep 1 14:25 hellojob.sh
-rw-rw-r-- 1 ec2-user ec2-user 43 Sep 1 14:30 slurm-2.out
```

Nell'output, vediamo un file out "». Possiamo vedere il risultato del nostro lavoro:

```
$ cat slurm-2.out
Hello World from queue1-dy-queue1t2micro-1
```

L'output mostra anche che il processo è stato eseguito correttamente sull'istanza queue1-dy-queue1t2micro-1.

Nel cluster appena creato, solo la home directory è condivisa tra tutti i nodi del cluster.

Per ulteriori informazioni sulla creazione e l'utilizzo dei cluster, consulta [Best practice](#).

Se l'applicazione richiede software, librerie o dati condivisi, considera le seguenti opzioni:

- Crea un'AMI personalizzata AWS ParallelCluster abilitata che includa il tuo software come descritto in [Creazione di un' AWS ParallelCluster AMI personalizzata](#).
- Utilizzate l'[StorageSettings](#) opzione nel file di AWS ParallelCluster configurazione per specificare un file system condiviso e archiviate il software installato nella posizione di montaggio specificata.
- [Azioni bootstrap personalizzate](#) Utilizzatelo per automatizzare la procedura di bootstrap di ogni nodo del cluster.

## Creazione di un' AWS ParallelCluster AMI personalizzata

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

### Important

Se crei un'AMI personalizzata, devi ripetere i passaggi utilizzati per creare l'AMI personalizzata con ogni nuova AWS ParallelCluster versione.

Prima di continuare a leggere, ti consigliamo di rivedere prima la [Azioni bootstrap personalizzate](#) sezione. Determina se le modifiche che desideri apportare possono essere scritte e supportate nelle versioni future AWS ParallelCluster .

Anche se la creazione di un'AMI personalizzata in generale non è l'ideale, esistono scenari specifici in cui AWS ParallelCluster è necessario creare un'AMI personalizzata per. Questo tutorial spiega come creare un'AMI personalizzata per questi scenari.

## Prerequisiti

- AWS ParallelCluster [è installato](#).
- AWS CLI [è installato e configurato](#).
- Hai una coppia di [chiavi Amazon EC2](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI e creare immagini.

## Come personalizzare l' AWS ParallelCluster AMI

Esistono due modi per creare un' AWS ParallelCluster AMI personalizzata. Uno di questi due metodi consiste nel creare una nuova AMI utilizzando la AWS ParallelCluster CLI. Un altro metodo richiede di apportare modifiche manuali per creare una nuova AMI disponibile con il tuo Account AWS.

## Crea un' AWS ParallelCluster AMI personalizzata

Se disponi di un'AMI e di un software personalizzati, puoi aggiungere AWS ParallelCluster le modifiche necessarie. AWS ParallelCluster si affida al servizio EC2 Image Builder per creare progetti personalizzati. AMIs Per ulteriori informazioni, consultare la Guida per l'[utente di Image Builder](#).

### Punti chiave:

- Il processo dura circa 1 ora. Questo tempo può variare se ci sono altri [Build/Components](#) da installare in fase di compilazione.
- L'AMI è contrassegnata con le versioni dei componenti principali. Questi includono il kernel, lo scheduler e il driver [EFA](#). Un sottoinsieme delle versioni dei componenti è riportato anche nella descrizione dell'AMI.
- A partire dalla AWS ParallelCluster versione 3.0.0, è possibile utilizzare un nuovo set di comandi CLI per gestire il ciclo di vita delle immagini. Sono inclusi [build-image](#), [list-images](#), [describe-image](#) e [delete-image](#).

- Questo metodo è ripetibile. È possibile rieseguirlo per rimanere AMIs aggiornato (ad esempio, aggiornamenti del sistema operativo) e quindi utilizzarlo quando si aggiorna un cluster esistente.

#### Note

Se si utilizza questo metodo nella partizione AWS cinese, è possibile che si verifichino errori di rete. Ad esempio, potresti visualizzare questi errori nel `pcluster build-image` comando quando scarica pacchetti da GitHub o da un archivio del sistema operativo. In tal caso, ti consigliamo di utilizzare uno dei seguenti metodi alternativi:

1. Segui l'[Modificare un AWS ParallelCluster AMI](#) approccio che ignora questo comando.
2. Crea l'immagine in un'altra partizione e regione, ad esempio `us-east-1`, e poi `store/restore` spostala nella regione della Cina. Per ulteriori informazioni, consulta [Archiviazione e ripristino di un'AMI utilizzando S3](#) nella Guida per l'utente di Amazon EC2.

Fasi:

1. Configura Account AWS le tue credenziali in modo che il AWS ParallelCluster client possa effettuare chiamate alle operazioni AWS API per tuo conto. Per un elenco delle autorizzazioni richieste, consulta [AWS Identity and Access Management autorizzazioni in AWS ParallelCluster](#).
2. Crea un file di configurazione dell'immagine di build di base. Per fare ciò, specifica il [InstanceType](#) file da utilizzare per creare l'immagine e il [ParentImage](#). Questi vengono utilizzati come punto di partenza per creare l'AMI. Per ulteriori informazioni sui parametri di compilazione opzionali, consulta [Image Configuration](#).

Build:

```
InstanceType: <BUILD_INSTANCE_TYPE>
ParentImage: <BASE_AMI_ID>
```

3. Usa il comando CLI `pcluster build-image` per creare un' AWS ParallelCluster AMI a partire dall'AMI che fornisci come base.

```
$ pcluster build-image --image-id IMAGE_ID --image-configuration IMAGE_CONFIG.yaml --
region REGION
{
 "image": {
 "imageId": "IMAGE_ID",
```

```
"imageBuildStatus": "BUILD_IN_PROGRESS",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
"region": "us-east-1",
"version": "3.15.0"
}
}
```

#### Warning

`pcluster build-image` utilizza il VPC predefinito. Se elimini il VPC predefinito utilizzando AWS Control Tower o AWS Landing Zone, l'ID della sottorete deve essere specificato nel file di configurazione dell'immagine. Per ulteriori informazioni, consulta [SubnetId](#).

Per un elenco di altri parametri, consulta la pagina di riferimento dei [pcluster build-image](#) comandi. I risultati del comando precedente sono i seguenti:

- Viene creato uno CloudFormation stack in base alla configurazione dell'immagine. Lo stack include tutte le risorse EC2 Image Builder necessarie per la compilazione.
  - Le risorse create includono i componenti ufficiali di Image Builder a cui è possibile aggiungere AWS ParallelCluster componenti Image Builder personalizzati. Per ulteriori informazioni, consulta [Creare un componente personalizzato con Image Builder](#) nella Guida per l'utente di EC2 Image Builder.
  - EC2 Image Builder avvia un'istanza di build, AWS ParallelCluster applica il ricettario, AWS ParallelCluster installa lo stack software ed esegue le attività di configurazione necessarie. Il AWS ParallelCluster ricettario viene utilizzato per creare e avviare. AWS ParallelCluster
  - L'istanza viene interrotta e da essa viene creata una nuova AMI.
  - Un'altra istanza viene lanciata dall'AMI appena creata. Durante la fase di test, EC2 Image Builder esegue test definiti nei componenti di Image Builder.
  - Se la compilazione ha esito positivo, lo stack viene eliminato. Se la compilazione fallisce, lo stack viene mantenuto e disponibile per l'ispezione.
4. È possibile monitorare lo stato del processo di compilazione eseguendo il comando seguente. Una volta completata la build, puoi eseguirla per recuperare l'ID AMI fornito nella risposta.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

```

BEFORE COMPLETE
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-
delete.s3.amazonaws.com/parallelcluster/3.15.0/images/IMAGE_ID-abcd1234efgh5678/
configs/image-config.yaml?...",
 },
 "imageId": "IMAGE_ID",
 "imagebuilderImageStatus": "BUILDING",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
 "region": "us-east-1",
 "version": "3.15.0",
 "cloudformationStackTags": [
 {
 "value": "3.15.0",
 "key": "parallelcluster:version"
 },
 {
 "value": "IMAGE_ID",
 "key": "parallelcluster:image_name"
 },
 ...
],
 "imageBuildLogsArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-IMAGE_ID",
 "cloudformationStackCreationTime": "2022-04-05T21:36:26.176Z"
}

AFTER COMPLETE
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.15.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
 },
 "imageId": "IMAGE_ID",
 "imageBuildStatus": "BUILD_COMPLETE",
 "region": "us-east-1",
 "ec2AmiInfo": {
 "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
 "amiId": "ami-1234stuv5678wxyz",

```

```

 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
 "state": "AVAILABLE",
 "tags": [
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 ...
],
 "architecture": "x86_64"
 },
 "version": "3.15.0"
}

```

5. Per creare il tuo cluster, inserisci l'ID AMI nel [CustomAmi](#) campo della configurazione del cluster.

## Risoluzione dei problemi e monitoraggio del processo di creazione di AMI

La creazione dell'immagine viene completata in circa un'ora. È possibile monitorare il processo eseguendo il [pcluster describe-image](#) comando o i comandi di recupero del registro.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

Il [build-image](#) comando crea uno CloudFormation stack con tutte le risorse Amazon EC2 necessarie per creare l'immagine e avvia il processo EC2 Image Builder.

Dopo aver eseguito il [build-image](#) comando, è possibile recuperare CloudFormation gli eventi dello stack utilizzando. [pcluster get-image-stack-events](#) Puoi filtrare i risultati con il `--query` parametro per vedere gli ultimi eventi. Per ulteriori informazioni, consulta [Filtraggio dell' AWS CLI output](#) nella Guida per l'AWS Command Line Interface utente.

```

$ pcluster get-image-stack-events --image-id IMAGE_ID --region REGION --query
"events[0]"
{
 "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
 "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-IMAGE_ID/3.15.0/1",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "Resource creation Initiated",

```

```

"resourceProperties": "{ \"InfrastructureConfigurationArn\":
 \"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
 parallelclusterimage-abcd1234-ef56-gh78-ij90-1234abcd5678\", \"ImageRecipeArn\":
 \"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/parallelclusterimage-
 IMAGE_ID/3.15.0\", \"DistributionConfigurationArn\": \"arn:aws:imagebuilder:us-
 east-1:123456789012:distribution-configuration/parallelclusterimage-abcd1234-ef56-
 gh78-ij90-1234abcd5678\", \"Tags\": { \"parallelcluster:image_name\": \"IMAGE_ID\",
 \"parallelcluster:image_id\": \"IMAGE_ID\" } }",
"stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/IMAGE_ID/abcd1234-
ef56-gh78-ij90-1234abcd5678",
"stackName": "IMAGE_ID",
"logicalResourceId": "ParallelClusterImage",
"resourceType": "AWS::ImageBuilder::Image",
"timestamp": "2022-04-05T21:39:24.725Z"
}

```

Dopo circa 15 minuti, gli eventi dello stack vengono visualizzati nella voce degli eventi di registro relativa alla creazione di Image Builder. È ora possibile elencare i flussi di log delle immagini e monitorare i passaggi di Image Builder [pcluster list-image-log-streams](#) utilizzando [pcluster get-image-log-events](#) i comandi and.

```

$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
 --query 'logStreams[*].logStreamName'

"3.15.0/1"
]

$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
 --log-stream-name 3.15.0/1 --limit 3
{
 "nextToken": "f/36295977202298886557255241372854078762600452615936671762",
 "prevToken": "b/36295977196879805474012299949460899222346900769983430672",
 "events": [
 {
 "message": "ExecuteBash: FINISHED EXECUTION",
 "timestamp": "2022-04-05T22:13:26.633Z"
 },
 {
 "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-abcd1234-ef56-gh78-ij90-1234abcd5678/3.15.0/1",
 "timestamp": "2022-04-05T22:13:26.741Z"
 },
 {

```

```

 "message": "TOE has completed execution successfully",
 "timestamp": "2022-04-05T22:13:26.819Z"
 }
]
}

```

Continuate a controllare con il [describe-image](#) comando finché non ne vedete lo stato.

**BUILD\_COMPLETE**

```

$ pcluster describe-image --image-id IMAGE_ID --region REGION
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-east-1.amazonaws.com/parallelcluster/3.15.0/images/IMAGE_ID-abcd1234efgh5678/configs/image-config.yaml?Signature=..."
 },
 "imageId": "IMAGE_ID",
 "imageBuildStatus": "BUILD_COMPLETE",
 "region": "us-east-1",
 "ec2AmiInfo": {
 "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
 "amiId": "ami-1234stuv5678wxyz",
 "description": "AWS ParallelCluster AMI for alinux2, kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64, efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
 "state": "AVAILABLE",
 "tags": [
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 ...
],
 "architecture": "x86_64"
 },
 "version": "3.15.0"
}

```

Se devi risolvere un problema di creazione di AMI personalizzate, crea un archivio dei log delle immagini come descritto nei passaggi seguenti.

È possibile archiviare i log in un bucket Amazon S3 o in un file locale, a seconda del parametro. --output



```
$ pcluster export-image-logs --image-id IMAGE_ID --region REGION \
--bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER \
{
 "url": "https://BUCKET_NAME.s3.us-east-1.amazonaws.com/BUCKET-FOLDER/IMAGE_ID-
logs-202209071136.tar.gz?AWSAccessKeyId=..."
}

$ pcluster export-image-logs --image-id IMAGE_ID \
--region REGION --bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER --output-file /tmp/
archive.tar.gz
{
 "path": "/tmp/archive.tar.gz"
}
```

L'archivio contiene i CloudWatch Logs Streams relativi al processo CloudFormation di Image Builder e agli eventi dello stack. L'esecuzione del comando potrebbe richiedere alcuni minuti.

## Gestione personalizzata AMIs

A partire dalla AWS ParallelCluster versione 3.0.0, è stato aggiunto un nuovo set di comandi nella CLI per creare, monitorare e gestire il ciclo di vita dell'immagine. [Per ulteriori informazioni sui comandi, consulta i comandi di pcluster.](#)

## Modificare un AWS ParallelCluster AMI

Questo metodo consiste nel modificare un' AWS ParallelCluster AMI ufficiale aggiungendovi delle personalizzazioni. La base AWS ParallelCluster AMIs viene aggiornata con nuove versioni. Questi AMIs hanno tutti i componenti necessari AWS ParallelCluster per funzionare una volta installati e configurati. Puoi iniziare con uno di questi come base.

### Punti chiave:

- Questo metodo è più veloce del [build-image](#) comando. Tuttavia, si tratta di un processo manuale e non ripetibile automaticamente.
- Con questo metodo, non è possibile accedere ai comandi di recupero dei log e di gestione del ciclo di vita delle immagini disponibili tramite la CLI.

### Fasi:

## New Amazon EC2 console

1. Trova l'AMI che corrisponde allo specifico Regione AWS che usi. Per trovarlo, usa il [pcluster list-official-images](#) comando con il `--region` parametro per selezionare i `--architecture` parametri Regione AWS `--os` e i parametri specifici da filtrare per l'AMI desiderata con il sistema operativo e l'architettura che desideri utilizzare. Dall'output, recupera l'ID immagine Amazon EC2.
2. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. <https://console.aws.amazon.com/ec2/>
3. Nel riquadro di navigazione, scegli Immagini, quindi AMIs. Cerca l'ID immagine EC2 recuperato, seleziona l'AMI e scegli Launch instance from AMI.
4. Scorri verso il basso e scegli il tipo di istanza.
5. Scegli la tua coppia di chiavi e Launch Instance.
6. Accedi alla tua istanza utilizzando l'utente del sistema operativo e la tua SSH chiave.
7. Personalizza manualmente l'istanza per soddisfare le tue esigenze.
8. Esegui il comando seguente per preparare l'istanza per la creazione di AMI.

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Dalla console, scegli Instance state e Stop instance.

Passa a Istanze, scegli la nuova istanza, seleziona Stato dell'istanza e Arresta istanza.

10. [Crea una nuova AMI dall'istanza utilizzando la console Amazon EC2 o AWS CLI create-image.](#)

Dalla console Amazon EC2

- a. Nel riquadro di navigazione scegliere Instances (Istanze).
  - b. Scegli l'istanza che hai creato e modificato.
  - c. In Azioni, scegli Immagine, quindi Crea immagine.
  - d. Scegliere Create Image (Crea immagine).
11. Inserisci il nuovo ID AMI nel [CustomAmi](#) campo della configurazione del cluster e crea un cluster.

## Old Amazon EC2 console

1. Trova l' AWS ParallelCluster AMI che corrisponde allo specifico Regione AWS che usi. Per trovarlo puoi usare il [pcluster list-official-images](#) comando con il `--region` parametro per selezionare i parametri specifici Regione AWS `--os` e `--architecture` i parametri da filtrare per l'AMI desiderata con il sistema operativo e l'architettura che desideri utilizzare. Dall'output puoi recuperare l'ID immagine di Amazon EC2.
2. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. <https://console.aws.amazon.com/ec2/>
3. Nel riquadro di navigazione, scegli Immagini, quindi AMIs. Imposta il filtro per le immagini pubbliche e cerca l'ID immagine EC2 recuperato, seleziona l'AMI e scegli Avvia.
4. Scegli il tipo di istanza e seleziona Avanti: Configura i dettagli dell'istanza o Rivedi e avvia per avviare l'istanza.
5. Scegli Launch, seleziona la tua coppia di chiavi e Launch Instances.
6. Accedere a un'istanza utilizzando l'utente del sistema operativo e la chiave SSH. Per ulteriori informazioni, accedi a Istanze, seleziona la nuova istanza e Connect.
7. Personalizza manualmente l'istanza per soddisfare le tue esigenze.
8. Eseguire il comando seguente per preparare l'istanza per la creazione di AMI:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Dalla console Amazon EC2, scegli Istanze nel riquadro di navigazione, seleziona la tua nuova istanza e scegli Actions, Instance State e Stop.
10. [Crea una nuova AMI dall'istanza utilizzando la console Amazon EC2 o AWS CLI create-image.](#)

Dalla console Amazon EC2

- a. Nel riquadro di navigazione scegliere Instances (Istanze).
  - b. Scegli l'istanza che hai creato e modificato.
  - c. In Azioni, scegliete Immagine, quindi Crea immagine.
  - d. Scegliere Create Image (Crea immagine).
11. Inserisci il nuovo ID AMI nel [CustomAmi](#) campo della configurazione del cluster e crea un cluster.

# Integrazione di Active Directory

In questo tutorial, crei un ambiente multiutente. Questo ambiente include un AWS ParallelCluster file integrato con un AWS Managed Microsoft AD (Active Directory) `incorp.example.com`. Si configura un Admin utente per gestire la directory, un ReadOnly utente per leggere la directory e un user000 utente per accedere al cluster. Puoi utilizzare il percorso automatico o il percorso manuale per creare le risorse di rete, un'Active Directory (AD) e l'istanza Amazon EC2 che usi per configurare l'AD. Indipendentemente dal percorso, l'infrastruttura creata è preconfigurata per AWS ParallelCluster l'integrazione utilizzando uno dei seguenti metodi:

- LDAPS con verifica del certificato (consigliato come opzione più sicura)
- LDAPS senza verifica del certificato
- LDAP

LDAP di per sé non fornisce la crittografia. Per garantire la trasmissione sicura di informazioni potenzialmente sensibili, consigliamo vivamente di utilizzare LDAPS (LDAP su TLS/SSL) per i cluster integrati con ADs. Per ulteriori informazioni, consulta [Abilitare l'utilizzo di LDAPS lato server nella Guida all'amministrazione](#). AWS Managed Microsoft AD Directory Service

Dopo aver creato queste risorse, procedi con la configurazione e la creazione del cluster integrato con Active Directory (AD). Dopo aver creato il cluster, accedi come utente che hai creato. Per ulteriori informazioni sulla configurazione creata in questo tutorial, consulta [Accesso di più utenti ai cluster](#) la sezione relativa alla [DirectoryService](#) configurazione.

Questo tutorial spiega come creare un ambiente che supporti l'accesso di più utenti ai cluster. Questo tutorial non spiega come creare e utilizzare un Directory Service AD. I passaggi da seguire per configurarne uno AWS Managed Microsoft AD in questo tutorial sono forniti solo a scopo di test. Non vengono forniti per sostituire la documentazione ufficiale e le best practice disponibili in [Simple AD](#) nella Directory Service Administration Guide. [AWS Managed Microsoft AD](#)

## Note

Le password degli utenti delle directory scadono in base alle definizioni delle proprietà delle policy relative alle password delle directory. Per reimpostare le password delle directory con AWS ParallelCluster, vedere. [Come reimpostare una password utente e le password scadute](#)

### Note

Gli indirizzi IP dei controller di dominio delle directory possono cambiare a causa delle modifiche dei controller di dominio e della manutenzione delle directory. Se si è scelto il metodo di creazione rapida automatizzata per creare l'infrastruttura di directory, è necessario allineare manualmente il sistema di bilanciamento del carico davanti ai controller delle directory quando gli indirizzi IP delle directory cambiano. Se si utilizza il metodo di creazione rapida, gli indirizzi IP delle directory non vengono allineati automaticamente ai sistemi di bilanciamento del carico.

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

### Prerequisiti

- AWS ParallelCluster [è](#) installato.
- AWS CLI [è installato e configurato](#).
- Hai una coppia di [chiavi Amazon EC2](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

Man mano che segui il tutorial, sostituisci *inputs highlighted in red*, ad esempio *region-id ed-abcdef01234567890*, con i tuoi nomi e. IDs Sostituisci *0123456789012* con il tuo Account AWS numero.

## Crea l'infrastruttura AD

Scegli la scheda Automatizzata per creare l'infrastruttura Active Directory (AD) con un modello di creazione CloudFormation rapida.

Scegli la scheda Manuale per creare manualmente l'infrastruttura AD.

### Automatizzata

1. Accedi alla Console di gestione AWS.
2. Apri [CloudFormation Quick Create \(region us-east-1\)](#) per creare le seguenti risorse nella console: CloudFormation

- Un VPC con due sottoreti e routing per l'accesso pubblico, se non viene specificato alcun VPC.
  - Un AWS Managed Microsoft AD.
  - Un'istanza Amazon EC2 aggiunta all'AD che puoi utilizzare per gestire la directory.
3. Nella sezione Parametri della pagina Quick create stack, inserisci le password per i seguenti parametri:
- AdminPassword
  - ReadOnlyPassword
  - UserPassword

Prendi nota delle password. Le userai più avanti in questo tutorial.

4. In DomainName, immetti **corp.example.com**.
5. Per Keypair, inserisci il nome di una coppia di chiavi Amazon EC2.
6. Seleziona le caselle per confermare ciascuna delle funzionalità di accesso nella parte inferiore della pagina.
7. Seleziona Crea stack.
8. Dopo che lo CloudFormation stack ha raggiunto lo CREATE\_COMPLETE stato, scegli la scheda Output dello stack. Prendi nota dei nomi delle risorse di output e IDs perché devi usarli nei passaggi successivi. Gli output forniscono le informazioni necessarie per creare il cluster.

CloudFormation > Stacks > PclusterAD-1234abc

**PclusterAD-abcd123** [Delete] [Update] [Stack actions] [Create stack]

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

**Outputs (10)**

| Key                               | Value                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------|
| DomainAddrLdap                    | ldap://10.0.111.88,ldap://10.0.222.111                                         |
| DomainAddrLdaps                   | ldaps://corp.example.com                                                       |
| DomainCertificateArn              | arn:aws:acm:us-east-1:123456789012:certificate/1234abcd-ef56-78gh-ij90-abcd1   |
| DomainCertificateSecretArn        | arn:aws:secretsmanager:us-east-1:123456789012:secret:DomainCertificateSecret-F |
| DomainCertificateSecretReadPolicy | arn:aws:iam::123456789012:policy/DomainCertificateSecretReadPolicy-PclusterAD  |
| DomainName                        | corp.example.com                                                               |
| DomainReadOnlyUser                | cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com                     |
| PasswordSecretArn                 | arn:aws:secretsmanager:us-east-1:123456789012:secret>PasswordSecret-PclusterA  |
| PrivateSubnetIds                  | subnet-1234567890abcdef0,subnet-abcdef01234567890                              |
| VpcId                             | vpc-021345abcdef6789                                                           |

9. Per completare gli esercizi [\(Facoltativo\) Gestisci utenti e gruppi AD](#), è necessario l'ID della directory. Scegli Risorse e scorri verso il basso per prendere nota dell'ID della directory.
10. Continua su [\(Facoltativo\) Gestisci utenti e gruppi AD](#) o [Creazione del cluster](#).

## Manuale

Crea un VPC per il servizio di directory con due sottoreti in diverse zone di disponibilità e un. AWS Managed Microsoft AD

## Crea l'AD

### Note

- La directory e il nome di dominio sono `corp.example.com`. Il nome breve è `CORP`.
- Cambia la Admin password nello script.
- La creazione di Active Directory (AD) richiede almeno 15 minuti.

Usa il seguente script Python per creare il VPC, le sottoreti e le risorse AD nel tuo locale. Regione AWS Salva questo file con nome `ed eseguiload.py`.

```
import boto3
import time
from pprint import pprint

vpc_name = "PclusterVPC"
ad_domain = "corp.example.com"
admin_password = "asdfASDF1234"

ec2 = boto3.client("ec2")
ds = boto3.client("ds")
region = boto3.Session().region_name

Create the VPC, Subnets, IGW, Routes
vpc = ec2.create_vpc(CidrBlock="10.0.0.0/16")["Vpc"]
vpc_id = vpc["VpcId"]
time.sleep(30)
ec2.create_tags(Resources=[vpc_id], Tags=[{"Key": "Name", "Value": vpc_name}])
subnet1 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.0.0/17",
 AvailabilityZone=f"{region}a")["Subnet"]
subnet1_id = subnet1["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet1_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/subnet1"}])
ec2.modify_subnet_attribute(SubnetId=subnet1_id, MapPublicIpOnLaunch={"Value": True})
subnet2 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.128.0/17",
 AvailabilityZone=f"{region}b")["Subnet"]
subnet2_id = subnet2["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet2_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/subnet2"}])
ec2.modify_subnet_attribute(SubnetId=subnet2_id, MapPublicIpOnLaunch={"Value": True})
igw = ec2.create_internet_gateway()["InternetGateway"]
ec2.attach_internet_gateway(InternetGatewayId=igw["InternetGatewayId"], VpcId=vpc_id)
route_table = ec2.describe_route_tables(Filters=[{"Name": "vpc-id", "Values":
 [vpc_id]}])["RouteTables"][0]
ec2.create_route(RouteTableId=route_table["RouteTableId"],
 DestinationCidrBlock="0.0.0.0/0", GatewayId=igw["InternetGatewayId"])
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsSupport={"Value": True})
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsHostnames={"Value": True})
```



```
Create the Active Directory
ad = ds.create_microsoft_ad(
 Name=ad_domain,
 Password=admin_password,
 Description="ParallelCluster AD",
 VpcSettings={"VpcId": vpc_id, "SubnetIds": [subnet1_id, subnet2_id]},
 Edition="Standard",
)
directory_id = ad["DirectoryId"]

Wait for completion
print("Waiting for the directory to be created...")
directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
directory = directories[0]
while directory["Stage"] in {"Requested", "Creating"}:
 time.sleep(3)
 directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
 directory = directories[0]

dns_ip_addrs = directory["DnsIpAddrs"]

pprint({"directory_id": directory_id,
 "vpc_id": vpc_id,
 "subnet1_id": subnet1_id,
 "subnet2_id": subnet2_id,
 "dns_ip_addrs": dns_ip_addrs})
```

Di seguito è riportato un esempio di output dallo script Python.

```
{
 "directory_id": "d-abcdef01234567890",
 "dns_ip_addrs": ["192.0.2.254", "203.0.113.237"],
 "subnet1_id": "subnet-021345abcdef6789",
 "subnet2_id": "subnet-1234567890abcdef0",
 "vpc_id": "vpc-021345abcdef6789"
}
```

Prendi nota dei nomi delle risorse di output e IDs. Li usi nei passaggi successivi.

Una volta completato lo script, vai al passaggio successivo.

## Creazione di un'istanza Amazon EC2

### New Amazon EC2 console

1. Accedi alla Console di gestione AWS.
2. Se non hai un ruolo con le politiche elencate nel passaggio 4 allegato, apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>. Altrimenti, vai al passaggio 5.
3. Crea la ResetUserPassword policy, sostituendo il contenuto evidenziato in rosso con il tuo Regione AWS ID, ID account e ID di directory dall'output dello script che hai eseguito per creare l'AD.

#### ResetUserPassword

```
{
 "Statement": [
 {
 "Action": [
 "ds:ResetUserPassword"
],
 "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
 "Effect": "Allow"
 }
]
}
```

4. Crea un ruolo IAM con le seguenti politiche allegate.
  - AWS politica gestita [Amazon SSMManaged InstanceCore](#)
  - AWS politica gestita [Amazon SSMDirectory ServiceAccess](#)
  - ResetUserPassword politica
5. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
6. Nella dashboard di Amazon EC2, scegli Launch Instance.
7. In Immagini dell'applicazione e del sistema operativo, seleziona un'AMI Amazon Linux 2 recente.
8. Per tipo di esempio, scegli t2.micro.
9. Per Coppia di chiavi, scegli una coppia di chiavi.
10. Per Impostazioni di rete, scegli Modifica.

11. Per VPC, seleziona la directory VPC.
12. Scorri verso il basso e seleziona Dettagli avanzati.
13. In Dettagli avanzati, Domain join directory, scegli **corp.example.com**.
14. Per il profilo IAM Instance, scegli il ruolo che hai creato nella fase 1 o un ruolo con le policy elencate nel passaggio 4 allegate.
15. In Riepilogo scegli Launch instance.
16. Prendi nota dell'ID dell'istanza (ad esempio, i-1234567890abcdef0) e attendi che l'istanza finisca di essere avviata.
17. Dopo l'avvio dell'istanza, procedi con il passaggio successivo.

## Old Amazon EC2 console

1. Accedi alla Console di gestione AWS.
2. Se non ricopri un ruolo con le politiche elencate nel passaggio 4 allegato, apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>. Altrimenti, vai al passaggio 5.
3. Crea la ResetUserPassword politica. Sostituisci il contenuto evidenziato in rosso con il tuo Regione AWS Account AWS ID, ID e l'ID di directory dall'output dello script che hai eseguito per creare Active Directory (AD).

### ResetUserPassword

```
{
 "Statement": [
 {
 "Action": [
 "ds:ResetUserPassword"
],
 "Resource": "arn:aws:ds:region-id:123456789012:directory/d-abcdef01234567890",
 "Effect": "Allow"
 }
]
}
```

4. Crea un ruolo IAM con le seguenti politiche allegate.
  - AWS politica gestita [Amazon SSMManaged InstanceCore](#)
  - AWS politica gestita [Amazon SSMDirectory ServiceAccess](#)

- Policy ResetUserPassword
5. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
  6. Nella dashboard di Amazon EC2, scegli Launch Instance.
  7. In Immagini dell'applicazione e del sistema operativo, seleziona un'AMI Amazon Linux 2 recente.
  8. In Instance type (Tipo di istanza) selezionare t2.micro.
  9. Per Coppia di chiavi, scegli una coppia di chiavi.
  10. In Impostazioni di rete scegli Modifica.
  11. In Impostazioni di rete, VPC, seleziona la directory VPC.
  12. Scorri verso il basso e seleziona Dettagli avanzati.
  13. In Dettagli avanzati, Domain join directory, scegli **corp.example.com**.
  14. In Dettagli avanzati, Profilo dell'istanza, scegli il ruolo creato nel passaggio 1 o un ruolo con le politiche elencate nel passaggio 4 allegate.
  15. In Riepilogo scegli Launch instance.
  16. Prendi nota dell'ID dell'istanza (ad esempio i-1234567890abcdef0) e attendi che l'istanza finisca di essere avviata.
  17. Dopo l'avvio dell'istanza, vai al passaggio successivo.

Unisci la tua istanza all'AD

1. Connettiti alla tua istanza e unisciti al regno AD come **admin**.

Esegui i seguenti comandi per connetterti all'istanza.

```
$ INSTANCE_ID="i-1234567890abcdef0"
```

```
$ PUBLIC_IP=$(aws ec2 describe-instances \
--instance-ids $INSTANCE_ID \
--query "Reservations[0].Instances[0].PublicIpAddress" \
--output text)
```

```
$ ssh -i ~/.ssh/keys/keypair.pem ec2-user@$PUBLIC_IP
```

2. Installa il software necessario ed entra a far parte del regno.

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-common-tools krb5-workstation openldap-clients policycoreutils-python
```

3. Sostituisci la password dell'amministratore con la tua **admin** password.

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com
Password for Admin:
```

Se la procedura precedente ha avuto successo, sei entrato a far parte del regno e puoi procedere al passaggio successivo.

## Aggiungi utenti all'AD

1. Crea il ReadOnlyUser e un altro utente.

In questo passaggio, si utilizzano gli strumenti [adcli](#) e [openldap-clients](#) installati in un passaggio precedente.

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=ReadOnlyUser ReadOnlyUser
```

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=user000 user000
```

2. Verifica che gli utenti siano stati creati:

Gli indirizzi IP DNS della directory sono output dello script Python.

```
$ DIRECTORY_IP="192.0.2.254"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=user000,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

Per impostazione predefinita, quando si crea un utente con `adcli`, l'utente è disabilitato.

3. Reimposta e attiva le password degli utenti dal computer locale:

Esci dalla tua istanza Amazon EC2.

#### Note

- `ro-p@ssw0rd` è la password di `ReadOnlyUser`, recuperata da Gestione dei segreti AWS
- `user-p@ssw0rd` è la password di un utente del cluster fornita quando ci si connette (ssh) al cluster.

`directory-id` È un output dello script Python.

```
$ DIRECTORY_ID="d-abcdef01234567890"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "ReadOnlyUser" \
--new-password "ro-p@ssw0rd" \
--region "region-id"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "user000" \
--new-password "user-p@ssw0rd" \
--region "region-id"
```

4. Aggiungi la password a un segreto di Secrets Manager.

Ora che hai creato `ReadOnlyUser` e impostato la password, memorizzala in un luogo segreto da AWS ParallelCluster utilizzare per la convalida degli accessi.

Usa Secrets Manager per creare un nuovo segreto che contenga la password `ReadOnlyUser` come valore. Il formato del valore segreto deve essere solo testo semplice (non in formato JSON). Prendi nota dell'ARN segreto per le fasi future.

```
$ aws secretsmanager create-secret --name "ADSecretPassword" \
--region region_id \
--secret-string "ro-p@ssw0rd" \
--query ARN \
--output text
arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
```

## Configurazione LDAPS con verifica del certificato (consigliata)

Prendi nota della risorsa IDs. Li usi nei passaggi successivi.

1. Genera un certificato di dominio, localmente.

```
$ PRIVATE_KEY="corp-example-com.key"
CERTIFICATE="corp-example-com.crt"
printf ".\n.\n.\n.\n.\ncorp.example.com\n.\n" | openssl req -x509 -sha256 -nodes -
newkey rsa:2048 -keyout $PRIVATE_KEY -days 365 -out $CERTIFICATE
```

2. Archivia il certificato in Secrets Manager per renderlo recuperabile dall'interno del cluster in un secondo momento.

```
$ aws secretsmanager create-secret --name example-cert \
--secret-string file://$CERTIFICATE \
--region region-id
{
 "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
 "Name": "example-cert",
 "VersionId": "14866070-092a-4d5a-bcdd-9219d0566b9c"
}
```

3. Aggiungi la seguente policy al ruolo IAM che hai creato per aggiungere l'istanza Amazon EC2 al dominio AD.

`PutDomainCertificateSecrets`

```
{
 "Statement": [
 {
 "Action": [
 "secretsmanager:PutSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc"
],
 "Effect": "Allow"
 }
]
}
```

4. Importa il certificato in AWS Certificate Manager (ACM).

```
$ aws acm import-certificate --certificate fileb://$CERTIFICATE \
 --private-key fileb://$PRIVATE_KEY \
 --region region-id
{
 "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
}
```

5. Crea e posiziona il load balancer davanti agli endpoint di Active Directory.

```
$ aws elbv2 create-load-balancer --name CorpExampleCom-NLB \
 --type network \
 --scheme internal \
 --subnets subnet-1234567890abcdef0 subnet-021345abcdef6789 \
 --region region-id
{
 "LoadBalancers": [
 {
 "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
 "CanonicalHostedZoneId": "Z2IFOLAFXWL04F",
 "CreatedTime": "2022-05-05T12:56:55.988000+00:00",
 "LoadBalancerName": "CorpExampleCom-NLB",
 "Scheme": "internal",

```



```

 "VpcId": "vpc-021345abcdef6789",
 "State": {
 "Code": "provisioning"
 },
 "Type": "network",
 "AvailabilityZones": [
 {
 "ZoneName": "region-idb",
 "SubnetId": "subnet-021345abcdef6789",
 "LoadBalancerAddresses": []
 },
 {
 "ZoneName": "region-ida",
 "SubnetId": "subnet-1234567890abcdef0",
 "LoadBalancerAddresses": []
 }
],
 "IpAddressType": "ipv4"
 }
]
}

```

## 6. Crea il gruppo target destinato agli endpoint di Active Directory.

```

$ aws elbv2 create-target-group --name CorpExampleCom-Targets --protocol TCP \
 --port 389 \
 --target-type ip \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id
{
 "TargetGroups": [
 {
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
 "TargetGroupName": "CorpExampleCom-Targets",
 "Protocol": "TCP",
 "Port": 389,
 "VpcId": "vpc-021345abcdef6789",
 "HealthCheckProtocol": "TCP",
 "HealthCheckPort": "traffic-port",
 "HealthCheckEnabled": true,
 "HealthCheckIntervalSeconds": 30,
 "HealthCheckTimeoutSeconds": 10,
 "HealthyThresholdCount": 3,

```

```

 "UnhealthyThresholdCount": 3,
 "TargetType": "ip",
 "IpAddressType": "ipv4"
 }
]
}

```

7. Registra gli endpoint Active Directory (AD) nel gruppo target.

```

$ aws elbv2 register-targets --target-group-arn
arn:aws:elasticloadbalancing:region-id:123456789012:targetgroup/CorpExampleCom-
Targets/44577c583b695e81 \
--targets Id=192.0.2.254,Port=389 Id=203.0.113.237,Port=389 \
--region region-id

```

8. Crea il listener LB con il certificato.

```

$ aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region-id:123456789012:loadbalancer/net/
CorpExampleCom-NLB/3afe296bf4ba80d4 \
--protocol TLS \
--port 636 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 \
--ssl-policy ELBSecurityPolicy-TLS-1-2-2017-01 \
--certificates CertificateArn=arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 \
--region region-id
"Listeners": [
{
 "ListenerArn": "arn:aws:elasticloadbalancing:region-id:123456789012:listener/
net/CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b",
 "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
 "Port": 636,
 "Protocol": "TLS",
 "Certificates": [
 {
 "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
 }
],
}
],

```

```

 "SslPolicy": "ELBSecurityPolicy-TLS-1-2-2017-01",
 "DefaultActions": [
 {
 "Type": "forward",
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
 "ForwardConfig": {
 "TargetGroups": [
 {
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81"
 }
]
 }
 }
]
 }
}

```

9. Crea la zona ospitata per rendere il dominio individuabile all'interno del VPC del cluster.

```

$ aws route53 create-hosted-zone --name corp.example.com \
 --vpc VPCRegion=region-id,VPCId=vpc-021345abcdef6789 \
 --caller-reference "ParallelCluster AD Tutorial"
{
 "Location": "https://route53.amazonaws.com/2013-04-01/hostedzone/
Z09020002B5MZQNXMSJUB",
 "HostedZone": {
 "Id": "/hostedzone/Z09020002B5MZQNXMSJUB",
 "Name": "corp.example.com.",
 "CallerReference": "ParallelCluster AD Tutorial",
 "Config": {
 "PrivateZone": true
 },
 "ResourceRecordSetCount": 2
 },
 "ChangeInfo": {
 "Id": "/change/C05533343BF3IKSORW1TQ",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T13:21:53.863000+00:00"
 },
 "VPC": {
 "VPCRegion": "region-id",

```

```

 "VPCId": "vpc-021345abcdef6789"
 }
}

```

10. Crea un file denominato **recordset-change.json** con il seguente contenuto.

**HostedZoneId** è l'ID canonico della zona ospitata del sistema di bilanciamento del carico.

```

{
 "Changes": [
 {
 "Action": "CREATE",
 "ResourceRecordSet": {
 "Name": "corp.example.com",
 "Type": "A",
 "Region": "region-id",
 "SetIdentifier": "example-active-directory",
 "AliasTarget": {
 "HostedZoneId": "Z2IF0LAFXWL04F",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
 "EvaluateTargetHealth": true
 }
 }
 }
]
}

```

11. Inviare la modifica del recordset alla zona ospitata, questa volta utilizzando l'ID della zona ospitata.

```

$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXMSJUB \
 --change-batch file://recordset-change.json
{
 "ChangeInfo": {
 "Id": "/change/C0137926I56R3GC7XW2Y",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
 }
}

```

12. Create un documento di policy **policy.json** con il seguente contenuto.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:us-east-1:123456789012:secret:example-cert-abc123"
],
 "Effect": "Allow"
 }
]
}
```

13. Create un documento di policy denominato **policy.json** con il seguente contenuto.

```
$ aws iam create-policy --policy-name ReadCertExample \
 --policy-document file://policy.json
{
 "Policy": {
 "PolicyName": "ReadCertExample",
 "PolicyId": "ANPAUUXUVBC42VZSI4LDY",
 "Arn": "arn:aws:iam::123456789012:policy/ReadCertExample-efg456",
 "Path": "/",
 "DefaultVersionId": "v1",
 "AttachmentCount": 0,
 "PermissionsBoundaryUsageCount": 0,
 "IsAttachable": true,
 "CreateDate": "2022-05-05T13:42:18+00:00",
 "UpdateDate": "2022-05-05T13:42:18+00:00"
 }
}
```

14. Continua a seguire i passaggi indicati in [\(Facoltativo\) Gestisci utenti e gruppi AD](#) o [Creazione del cluster](#).

## (Facoltativo) Gestisci utenti e gruppi AD

In questa fase, gestisci utenti e gruppi da un'istanza Amazon EC2 Amazon Linux 2 aggiunta al dominio Active Directory (AD).

Se hai seguito il percorso automatizzato, riavvia e accedi all'istanza unita AD creata come parte dell'automazione.

Se hai seguito il percorso manuale, riavvia e accedi all'istanza che hai creato e che hai aggiunto all'AD nei passaggi precedenti.

In questi passaggi, si utilizzano gli strumenti [adcli](#) e [openldap-clients](#) installati nell'istanza come parte del passaggio precedente.

Accedi a un'istanza Amazon EC2 che fa parte del dominio AD

1. Dalla console Amazon EC2, seleziona l'istanza Amazon EC2 senza titolo creata nei passaggi precedenti. Lo stato dell'istanza potrebbe essere Interrotto.
2. Se lo stato dell'istanza è Interrotto, scegli Stato dell'istanza e quindi Avvia istanza.
3. Una volta superati i controlli di stato, seleziona l'istanza e scegli Connect e accedi SSH all'istanza.

Gestisci utenti e gruppi quando accedi a un'istanza Amazon EC2 Amazon Linux 2 che fa parte di AD

Quando `adcli` esegui i comandi con l' `-U "Admin"` opzione, ti viene richiesto di inserire la password AD. Admin Include la Admin password AD come parte dei `ldapsearch` comandi.

1. Creare un utente.

```
$ adcli create-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

2. Imposta una password utente.

```
$ aws --region "region-id" ds reset-user-password --directory-id "d-
abcdef01234567890" --user-name "clusteruser" --new-password "new-p@ssw0rd"
```

3. Crea un gruppo.

```
$ adcli create-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

#### 4. Aggiungi un utente a un gruppo.

```
$ adcli add-member "clusterteam" "clusteruser" --domain "corp.example.com" -U "Admin"
```

#### 5. Descrivi utenti e gruppi.

Descrivi tutti gli utenti.

```
$ ldapsearch "(&(objectClass=user))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Descrivi un utente specifico.

```
$ ldapsearch "(&(objectClass=user)(cn=clusteruser))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Descrivi tutti gli utenti con uno schema di nomi.

```
$ ldapsearch "(&(objectClass=user)(cn=user*))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Descrivi tutti gli utenti che fanno parte di un gruppo specifico.

```
$ ldapsearch "(&(objectClass=user)(memberOf=CN=clusterteam,OU=Users,OU=CORP,DC=corp,DC=example,DC=com))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Descrivi tutti i gruppi

```
$ ldapsearch "objectClass=group" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Descrivi un gruppo specifico

```
$ ldapsearch "(&(objectClass=group)(cn=clusterteam))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

6. Rimuovi un utente da un gruppo.

```
$ adcli remove-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

7. Elimina un utente.

```
$ adcli delete-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

8. Eliminare un gruppo.

```
$ adcli delete-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

## Creazione del cluster

Se non sei ancora uscito dall'istanza Amazon EC2, fallo ora.

L'ambiente è configurato per creare un cluster in grado di autenticare gli utenti tramite Active Directory (AD).

Crea una semplice configurazione del cluster e fornisci le impostazioni relative alla connessione ad AD. Per ulteriori informazioni, consulta la sezione [DirectoryService](#).

Scegli una delle seguenti configurazioni del cluster e copiala in un file denominato `ldaps_config.yaml`, `ldaps_nocert_config.yaml`, `oldap_config.yaml`.

Ti consigliamo di scegliere la configurazione LDAPS con verifica del certificato. Se scegli questa configurazione, devi anche copiare lo script di bootstrap in un file denominato `active-directory.head.post.sh`. Inoltre, è necessario archivarlo in un bucket Amazon S3 come indicato nel file di configurazione.



## LDAPS con configurazione di verifica del certificato (consigliato)

### Note

I seguenti componenti devono essere modificati.

- **KeyName**: Una delle tue coppie di chiavi Amazon EC2.
- **SubnetId / SubnetIds**: Una delle sottoreti IDs fornite nell'output dello stack di creazione CloudFormation rapida (tutorial automatico) o dello script python (tutorial manuale).
- **Region**: La regione in cui hai creato l'infrastruttura AD.
- **DomainAddr**: Questo indirizzo IP è uno degli indirizzi DNS del servizio AD.
- **PasswordSecretArn**: L'Amazon Resource Name (ARN) del segreto che contiene la password per `DomainReadOnlyUser`.
- **BucketName**: il nome del bucket che contiene lo script di bootstrap.
- **AdditionalPolicies/Policy**: L'Amazon Resource Name (ARN) della politica di certificazione del dominio di lettura. `ReadCertExample`.
- **CustomActions/OnNodeConfigured/Args**: L'Amazon Resource Name (ARN) segreto che contiene la politica di certificazione del dominio.

Per una migliore sicurezza, suggeriamo di utilizzare la `AllowedIps` configurazione `HeadNode/Ssh` per limitare l'accesso SSH al nodo principale.

Si noti che il certificato specificato in `LdapTlsCaCert` deve essere accessibile a tutti i nodi del cluster.

### Requisiti rigidi

- Il certificato specificato in `LdapTlsCaCert` deve essere accessibile a tutti i nodi del cluster.

Un nodo senza accesso al certificato non sarà in grado di risolvere gli utenti dalla directory.

Region: *region-id*

```
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ReadCertExample
 S3Access:
 - BucketName: amzn-s3-demo-bucket
 EnableWriteAccess: false
 KeyName: bootstrap/active-directory/active-directory.head.post.sh
 CustomActions:
 OnNodeConfigured:
 Script: s3://amzn-s3-demo-bucket/bootstrap/active-directory/active-
directory.head.post.sh
 Args:
 - arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc
 - /opt/parallelcluster/shared/directory_service/domain-certificate.crt
 Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
 DirectoryService:
 DomainName: corp.example.com
 DomainAddr: ldaps://corp.example.com
 PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 LdapTlsCaCert: /opt/parallelcluster/shared/directory_service/domain-certificate.crt
 LdapTlsReqCert: hard
```

## Script Bootstrap

Dopo aver creato il file bootstrap e prima di caricarlo nel bucket S3, eseguirlo per concedere l'autorizzazione `chmod +x active-directory.head.post.sh` all'esecuzione. AWS ParallelCluster

```
#!/bin/bash
set -e

CERTIFICATE_SECRET_ARN="$1"
CERTIFICATE_PATH="$2"

[[-z $CERTIFICATE_SECRET_ARN]] && echo "[ERROR] Missing CERTIFICATE_SECRET_ARN" &&
exit 1
[[-z $CERTIFICATE_PATH]] && echo "[ERROR] Missing CERTIFICATE_PATH" && exit 1

source /etc/parallelcluster/cfnconfig
REGION="${cfn_region:?}"

mkdir -p $(dirname $CERTIFICATE_PATH)
aws secretsmanager get-secret-value --region $REGION --secret-id
$CERTIFICATE_SECRET_ARN --query SecretString --output text > $CERTIFICATE_PATH
```

## Configurazione LDAPS senza verifica del certificato

### Note

I seguenti componenti devono essere modificati.

- **KeyName:** Una delle tue coppie di chiavi Amazon EC2.
- **SubnetId / SubnetIds:** Una delle sottoreti IDs presenti nell'output dello stack di creazione CloudFormation rapida (tutorial automatizzato) o dello script python (tutorial manuale).
- **Region:** La regione in cui hai creato l'infrastruttura AD.
- **DomainAddr:** Questo indirizzo IP è uno degli indirizzi DNS del servizio AD.
- **PasswordSecretArn:** L'Amazon Resource Name (ARN) del segreto che contiene la password per. DomainReadOnlyUser

Per una migliore sicurezza, suggeriamo di utilizzare la AllowedIps configurazione HeadNode / Ssh/ per limitare l'accesso SSH al nodo principale.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: corp.example.com
 DomainAddr: ldaps://corp.example.com
 PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 LdapTlsReqCert: never
```

## Configurazione LDAP

### Note

I seguenti componenti devono essere modificati.

- KeyName: Una delle tue coppie di chiavi Amazon EC2.

- **SubnetId / SubnetIds:** Una delle sottoreti IDs fornite nell'output dello stack di creazione CloudFormation rapida (tutorial automatico) o dello script python (tutorial manuale).
- **Region:** La regione in cui hai creato l'infrastruttura AD.
- **DomainAddr:** Questo indirizzo IP è uno degli indirizzi DNS del servizio AD.
- **PasswordSecretArn:** L'Amazon Resource Name (ARN) del segreto che contiene la password per. DomainReadOnlyUser

Per una migliore sicurezza, suggeriamo di utilizzare la AllowedIps configurazione HeadNode / Ssh/ per limitare l'accesso SSH al nodo principale.

```

Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: dc=corp,dc=example,dc=com
 DomainAddr: ldap://192.0.2.254,ldap://203.0.113.237
 PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 AdditionalSssdConfigs:

```

```
ldap_auth_disable_tls_never_use_in_production: True
```

Crea un cluster con il seguente comando.

```
$ pcluster create-cluster --cluster-name "ad-cluster" --cluster-configuration "./
ldaps_config.yaml"
{
 "cluster": {
 "clusterName": "pcluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-
cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "region-id",
 "version": 3.15.0,
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

## Connect al cluster come utente

È possibile determinare lo stato del cluster con i seguenti comandi.

```
$ pcluster describe-cluster -n ad-cluster --region "region-id" --query "clusterStatus"
```

L'output è il seguente.

```
"CREATE_IN_PROGRESS" / "CREATE_COMPLETE"
```

Quando lo stato viene raggiunto "CREATE\_COMPLETE", accedi con il nome utente e la password creati.

```
$ HEAD_NODE_IP=$(pcluster describe-cluster -n "ad-cluster" --region "region-id" --query
headNode.publicIpAddress | xargs echo)
```

```
$ ssh user000@$HEAD_NODE_IP
```

È possibile accedere senza la password fornendo la SSH chiave creata per il nuovo utente all'indirizzo `/home/user000@$HEAD_NODE_IP/.ssh/id_rsa`.

Se il `ssh` comando ha avuto esito positivo, la connessione al cluster è avvenuta correttamente come utente autenticato per l'utilizzo di Active Directory (AD).

## Eliminazione

1. Dal computer locale, elimina il cluster.

```
$ pcluster delete-cluster --cluster-name "ad-cluster" --region "region-id"
{
 "cluster": {
 "clusterName": "ad-cluster",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "region-id",
 "version": "3.15.0",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

2. Controlla lo stato di avanzamento del cluster da eliminare.

```
$ pcluster describe-cluster --cluster-name "ad-cluster" --region "region-id" --
query "clusterStatus"
"DELETE_IN_PROGRESS"
```

Dopo che il cluster è stato eliminato con successo, procedi al passaggio successivo.

### Automatizzato

#### Eliminare le risorse di Active Directory

1. Da <https://console.aws.amazon.com/cloudformation/>.
2. Nel riquadro di navigazione selezionare Stacks (Stack).
3. Dall'elenco degli stack, scegli lo stack AD (ad esempio, `pcluster-ad`).
4. Scegli Elimina.

## Manuale

1. Elimina l'istanza Amazon EC2.
  - a. Da <https://console.aws.amazon.com/ec2/>, scegli Istanze nel riquadro di navigazione.
  - b. Dall'elenco delle istanze, scegli l'istanza che hai creato per aggiungere utenti alla directory.
  - c. Scegli Stato dell'istanza, quindi Termina istanza.
2. Elimina la zona ospitata.
  - a. Crea un `recordset-delete.json` file con il seguente contenuto. In questo esempio, `HostedZoneId` è l'ID canonico della zona ospitata del load balancer.

```
{
 "Changes": [
 {
 "Action": "DELETE",
 "ResourceRecordSet": {
 "Name": "corp.example.com",
 "Type": "A",
 "Region": "region-id",
 "SetIdentifier": "pcluster-active-directory",
 "AliasTarget": {
 "HostedZoneId": "Z2IF0LAFXWL04F",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
 "EvaluateTargetHealth": true
 }
 }
 }
]
}
```

- b. Inviare la modifica del recordset alla zona ospitata utilizzando l'ID della zona ospitata.

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXSJUB \
 --change-batch file://recordset-delete.json
{
 "ChangeInfo": {
 "Id": "/change/C04853642A0TH2TJ5NLNI",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T14:25:51.046000+00:00"
```



```
}
}
```

- c. Eliminare la zona ospitata.

```
$ aws route53 delete-hosted-zone --id Z09020002B5MZQNXMSJUB
{
 "ChangeInfo": {
 "Id": "/change/C0468051QFABTVHMDEG9",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T14:26:13.814000+00:00"
 }
}
```

3. Eliminare il listener LB.

```
$ aws elbv2 delete-listener \
 --listener-arn arn:aws:elasticloadbalancing:region-id:123456789012:listener/net/
 CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b --region region-id
```

4. Eliminare il gruppo target.

```
$ aws elbv2 delete-target-group \
 --target-group-arn arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 --
 region region-id
```

5. Eliminare il sistema di bilanciamento del carico.

```
$ aws elbv2 delete-load-balancer \
 --load-balancer-arn arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4 --
 region region-id
```

6. Eliminare la politica utilizzata dal cluster per leggere il certificato da Secrets Manager.

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/
ReadCertExample
```

7. Elimina il segreto che contiene il certificato di dominio.

```
$ aws secretsmanager delete-secret \

```

```
--secret-id arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc \
--region region-id
{
 "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",
 "Name": "example-cert",
 "DeletionDate": "2022-06-04T16:27:36.183000+02:00"
}
```

8. Eliminare il certificato da ACM.

```
$ aws acm delete-certificate \
--certificate-arn arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 --region region-id
```

9. Eliminare le risorse di Active Directory (AD).

a. Ottieni la seguente risorsa IDs dall'output dello script `ad.py` python:

- E ID
- Sottorete AD IDs
- E ID VPC

b. Eliminare la directory eseguendo il comando seguente.

```
$ aws ds delete-directory --directory-id d-abcdef0123456789 --region region-id
{
 "DirectoryId": "d-abcdef0123456789"
}
```

c. Elenca i gruppi di sicurezza nel VPC.

```
$ aws ec2 describe-security-groups --filters '[{"Name":"vpc-id","Values":
["vpc-07614ade95ebad1bc"]}]' --region region-id
```

d. Elimina il gruppo di sicurezza personalizzato.

```
$ aws ec2 delete-security-group --group-id sg-021345abcdef6789 --region region-
id
```

e. Elimina le sottoreti.

```
$ aws ec2 delete-subnet --subnet-id subnet-1234567890abcdef --region region-id
```

```
$ aws ec2 delete-subnet --subnet-id subnet-021345abcdef6789 --region region-id
```

f. Descrivi Internet Gateway.

```
$ aws ec2 describe-internet-gateways \
 --filters Name=attachment.vpc-id,Values=vpc-021345abcdef6789 \
 --region region-id
{
 "InternetGateways": [
 {
 "Attachments": [
 {
 "State": "available",
 "VpcId": "vpc-021345abcdef6789"
 }
],
 "InternetGatewayId": "igw-1234567890abcdef",
 "OwnerId": "123456789012",
 "Tags": []
 }
]
}
```

g. Scollegare il gateway Internet.

```
$ aws ec2 detach-internet-gateway \
 --internet-gateway-id igw-1234567890abcdef \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id
```

h. Eliminare il gateway Internet.

```
$ aws ec2 delete-internet-gateway \
 --internet-gateway-id igw-1234567890abcdef \
 --region region-id
```

i. Elimina il VPC.

```
$ aws ec2 delete-vpc \
```

```
--vpc-id vpc-021345abcdef6789 \
--region region-id
```

- j. Eliminare il segreto che contiene la ReadOnlyUser password.

```
$ aws secretsmanager delete-secret \
 --secret-id arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234" \
 --region region-id
```

## Configurazione della crittografia dello storage condiviso con una chiave AWS KMS

Scopri come configurare una AWS KMS chiave gestita dal cliente per crittografare e proteggere i dati nei sistemi di archiviazione di file in cluster per cui sono configurati. AWS ParallelCluster

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

AWS ParallelCluster supporta le seguenti opzioni di configurazione dello storage condiviso:

- [SharedStorage](#) / [EbsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [EfsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [FsxLustreSettings](#) / [KmsKeyId](#)

Puoi utilizzare queste opzioni per fornire una AWS KMS chiave gestita dal cliente per Amazon EBS, Amazon EFS e FSx per la crittografia del sistema di storage condiviso Lustre. Per utilizzarle, devi creare e configurare una policy IAM per quanto segue:

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

### Prerequisiti

- AWS ParallelCluster [è installato](#).
- AWS CLI [è installato e configurato](#).

- Hai una coppia di [chiavi Amazon EC2](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

## Argomenti

- [Creare la policy](#)
- [Configura e crea il cluster](#)

## Creare la policy

In questo tutorial, creerai una politica per configurare la crittografia dello storage condiviso con una AWS KMS chiave.

Creazione di una policy.

1. Vai alla console IAM: <https://console.aws.amazon.com/iam/home>.
2. Seleziona Policy.
3. Scegli Crea policy.
4. Scegli la scheda JSON e incolla la seguente policy. Assicurati di sostituire tutte le occorrenze di **123456789012** con il tuo Account AWS ID e la chiave Amazon Resource Name (ARN) Regione AWS e con il tuo.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:us-east-1:123456789012:key/abcd1234-ef56-gh78-ij90-abcd1234efgh5678"
]
 }
]
}
```

```
 }
]
}
```

5. Per questo tutorial, assegna un nome alla politica `ParallelClusterKmsPolicy`, quindi scegli Crea politica.
6. Prendi nota della politica ARN. Ne hai bisogno per configurare il tuo cluster.

## Configura e crea il cluster

Di seguito è riportato un esempio di configurazione del cluster che include un file system condiviso di Amazon Elastic Block Store con crittografia.

```
Region: eu-west-1
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: my-ssh-key
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: q1
 ComputeResources:
 - Name: t2micro
 InstanceType: t2.micro
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
SharedStorage:
```

```
- MountDir: /shared/efs1
 Name: shared-efs1
 StorageType: Ebs
 EbsSettings:
 Encrypted: True
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

Sostituisci gli elementi in rosso con i tuoi valori. Quindi, crea un cluster che utilizza la tua AWS KMS chiave per crittografare i dati in Amazon EBS.

La configurazione è simile per Amazon EFS e FSx per i file system Lustre.

La SharedStorage configurazione di Amazon EFS è la seguente.

```
...
SharedStorage:
 - MountDir: /shared/efs1
 Name: shared-efs1
 StorageType: Efs
 EfsSettings:
 Encrypted: True
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

La SharedStorage configurazione di FSx for Lustre è la seguente.

```
...
SharedStorage:
 - MountDir: /shared/fsx1
 Name: shared-fsx1
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeploymentType: PERSISTENT_1
 PerUnitStorageThroughput: 200
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

## Esecuzione di processi in un cluster in modalità coda multipla

Questo tutorial spiega come eseguire il primo lavoro "Hello World" in AWS ParallelCluster modalità [coda multipla](#).

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

## Prerequisiti

- AWS ParallelCluster [è](#) installato.
- AWS CLI [è installato e configurato](#).
- Hai una coppia di [chiavi Amazon EC2](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

## Configura il tuo cluster

Innanzitutto, verifica che AWS ParallelCluster sia installato correttamente eseguendo il comando seguente.

```
$ pcluster version
```

Per ulteriori informazioni su `pcluster version`, consultare [pcluster version](#).

Questo comando restituisce la versione in esecuzione di AWS ParallelCluster.

Quindi, `pcluster configure` esegui per generare un file di configurazione di base. Segui tutte le istruzioni che seguono questo comando.

```
$ pcluster configure --config multi-queue-mode.yaml
```

Per ulteriori informazioni sul comando `pcluster configure`, consulta [pcluster configure](#).

Dopo aver completato questo passaggio, viene visualizzato un file di configurazione di base denominato `multi-queue-mode.yaml`. Questo file contiene una configurazione di base del cluster.

Nel passaggio successivo, si modifica il nuovo file di configurazione e si avvia un cluster con più code.

### Note

Alcune istanze utilizzate in questo tutorial non sono idonee al livello gratuito.



Per questo tutorial, modifica il file di configurazione in modo che corrisponda alla configurazione seguente. Gli elementi evidenziati in rosso rappresentano i valori del file di configurazione. Mantieni i tuoi valori.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: c5.xlarge
Networking:
 SubnetId: subnet-abcdef01234567890
Ssh:
 KeyName: yourkeypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: spot
 ComputeResources:
 - Name: c5xlarge
 InstanceType: c5.xlarge
 MinCount: 1
 MaxCount: 10
 - Name: t2micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
 - Name: ondemand
 ComputeResources:
 - Name: c52xlarge
 InstanceType: c5.2xlarge
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-021345abcdef6789
```

## Creazione di un cluster

Crea un cluster denominato `multi-queue-cluster` base al tuo file di configurazione.

```
$ pcluster create-cluster --cluster-name multi-queue-cluster --cluster-configuration
multi-queue-mode.yaml
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.15.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Per ulteriori informazioni sul comando `pcluster create-cluster`, consulta [pcluster create-cluster](#).

Per verificare lo stato del cluster, esegui il comando seguente.

```
$ pcluster list-clusters
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.15.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Quando il cluster viene creato, il `clusterStatus` campo viene visualizzato `CREATE_COMPLETE`.

## Accedere al nodo principale

Usa il tuo file di chiave SSH privato per accedere al nodo principale.

```
$ pcluster ssh --cluster-name multi-queue-cluster -i ~/path/to/yourkeyfile.pem
```

Per ulteriori informazioni su `pcluster ssh`, consultare [pcluster ssh](#).

Dopo aver effettuato l'accesso, esegui il `sinfo` comando per verificare che le code dello scheduler siano impostate e configurate.

Per maggiori informazioni in merito `sinfo`, consulta [sinfo](#) nella documentazione. Slurm

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 18 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[1-9]
spot* up infinite 2 idle spot-st-c5xlarge-1,spot-st-t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

L'output mostra che hai uno `t2.micro` e un nodo di `c5.xlarge` calcolo nello `idle` stato disponibile nel tuo cluster.

Gli altri nodi sono tutti nello stato di risparmio energetico, indicato dal `~` suffisso nello stato del nodo, senza istanze Amazon EC2 che li supportano. La coda predefinita è indicata da un `*` suffisso dopo il nome della coda. `spot` è la tua coda di lavoro predefinita.

## Esegue il lavoro in modalità coda multipla

Quindi, prova a eseguire un lavoro per dormire per qualche istante. Successivamente, il job restituisce il proprio nome host. Assicuratevi che questo script possa essere eseguito dall'utente corrente.

```
$ tee <<EOF hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from \$(hostname)"
EOF

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

Inviare il lavoro utilizzando il `sbatch` comando. Richiedi due nodi per questo lavoro con l' `-N 2` opzione e verifica che il lavoro sia stato inviato correttamente. Per ulteriori informazioni [sbatch](#) in `meritosbatch`, consulta la documentazione di Slurm.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
```

```
Submitted batch job 1
```

È possibile visualizzare la coda e controllare lo stato del lavoro con il comando `squeue`. Poiché non è stata specificata una coda specifica, viene utilizzata la coda predefinita (`spot`). Per ulteriori informazioni in merito a `squeue`, consulta la [squeue](#) Slurmdocumentazione.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 1 spot wrap ec2-user R 0:10 2 spot-st-c5xlarge-1,spot-st-
t2micro-1
```

L'output indica che il processo è attualmente in un stato di esecuzione. Attendi che il lavoro finisca. Questa operazione richiede circa 30 secondi. Quindi, corri `squeue` di nuovo.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Ora che tutti i lavori in coda sono terminati, cercate il file di output denominato `slurm-1.out` nella directory corrente.

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```

L'output mostra che il processo è stato eseguito correttamente sui `spot-st-c5xlarge-1` nodi `spot-st-t2micro-1` and.

Ora inviate lo stesso lavoro specificando i vincoli per istanze specifiche con i seguenti comandi.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 2
```

Hai usato questi parametri per: `sbatch`

- `-N 3`— richiede tre nodi.
- `-p spot`— invia il lavoro alla `spot` coda. È inoltre possibile inviare un lavoro alla `ondemand` coda specificando. `-p ondemand`
- `-C "[c5.xlarge*1&t2.micro*2]"`— specifica i vincoli di nodo specifici per questo lavoro. Ciò richiede l'utilizzo di un `c5.xlarge` nodo e due `t2.micro` nodi per questo lavoro.

Esegui il `sinfo` comando per visualizzare i nodi e le code. Le code in entrata AWS ParallelCluster sono chiamate partizioni in. Slurm

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 1 alloc# spot-dy-t2micro-1
spot* up infinite 17 idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot* up infinite 1 mix spot-st-c5xlarge-1
spot* up infinite 1 alloc spot-st-t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

I nodi si stanno accendendo. Ciò è indicato dal `#` suffisso sullo stato del nodo. Esegui il `squeue` comando per visualizzare le informazioni sui lavori nel cluster.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 spot wrap ec2-user CF 0:04 3 spot-dy-c5xlarge-1,spot-dy-
t2micro-1,spot-st-t2micro-1
```

Il processo è nello stato CF (CONFIGURING), in attesa che le istanze aumentino di dimensioni e si uniscano al cluster.

Dopo circa tre minuti, i nodi sono disponibili e il processo entra nello stato R (RUNNING).

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 spot wrap ec2-user R 0:07 3 spot-dy-t2micro-1,spot-st-
c5xlarge-1,spot-st-t2micro-1
```

Il processo termina e tutti e tre i nodi sono nello `idle` stato.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 17 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot* up infinite 3 idle spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Quindi, dopo che nessun lavoro rimane in coda, cercate `slurm-2.out` nella vostra rubrica locale.

```
$ cat slurm-2.out
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
Hello World from spot-st-c5xlarge-1
```

Questo è lo stato finale del cluster.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 17 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot* up infinite 3 idle spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Dopo esserti disconnesso dal cluster, puoi ripulirlo `pcluster delete-cluster` eseguendo. Per ulteriori informazioni, consultare [pcluster list-clusters](#) e [pcluster delete-cluster](#).

```
$ pcluster list-clusters
{
 "clusters": [
 {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_COMPLETE"
 }
]
}

$ pcluster delete-cluster -n multi-queue-cluster
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.1.4",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

```
}
```

## Utilizzo dell' AWS ParallelCluster API

In questo tutorial, creerai e testerai l'API con [Amazon API Gateway](#) e un AWS ParallelCluster CloudFormation modello. Quindi, usi il client di esempio disponibile su GitHub per utilizzare l'API. Per ulteriori informazioni sull'uso dell'API, consulta [AWS ParallelCluster API](#).

Per ulteriori informazioni, consulta [Creare un componente personalizzato con Image Builder](#) nella Guida per l'utente di Image EC2 Builder.

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

### Prerequisiti

- AWS CLI Viene [installato](#) e configurato nel tuo ambiente di elaborazione.
- AWS ParallelCluster è installato in un ambiente virtuale. Per ulteriori informazioni, consulta [Installa AWS ParallelCluster in un ambiente virtuale \(consigliato\)](#).
- Hai una [coppia di EC2 chiavi Amazon](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

## Fase 1: Crea l'API con Amazon API Gateway

Rimani nella tua directory utente principale e attiva il tuo ambiente virtuale:

1. Installa un utile processore a riga di comando JSON.

```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. Esegui il comando seguente per ottenere la tua AWS ParallelCluster versione e assegnarla a una variabile di ambiente.

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" | tee -a ~/.bashrc
```

3. Crea una variabile di ambiente e assegna il tuo ID regionale.

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" |tee -a ~/.bashrc
```

4. Esegui i seguenti comandi per distribuire l'API.

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" |tee -a ~/.bashrc
```

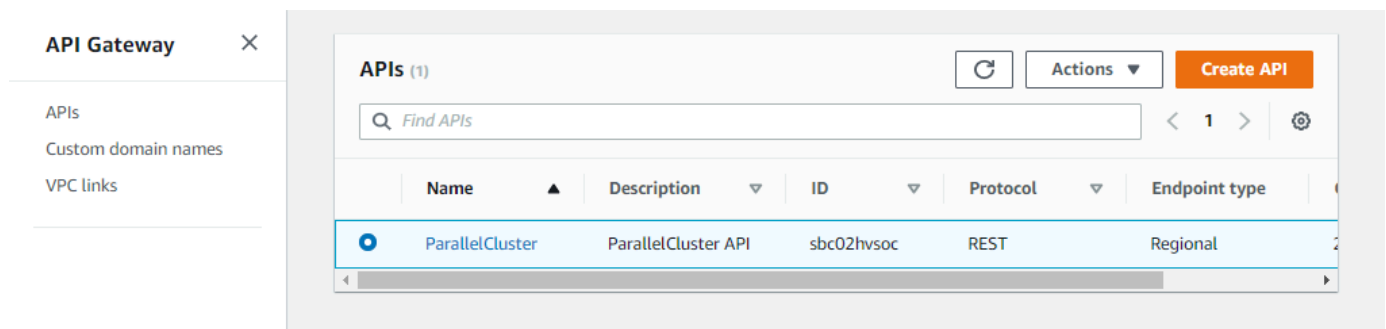
```
aws cloudformation create-stack \
 --region ${AWS_DEFAULT_REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
 --parameters ParameterKey=EnableIamAdminAccess,ParameterValue=true

{
 "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-api-
stack/abcd1234-ef56-gh78-ei90-1234abcd5678"
}
```

Al termine del processo, procedi al passaggio successivo.

## Fase 2: Verifica l'API nella console Amazon API Gateway

1. Accedi alla Console di gestione AWS.
2. Accedi alla [console Amazon API Gateway](#).
3. Scegli la tua implementazione API.

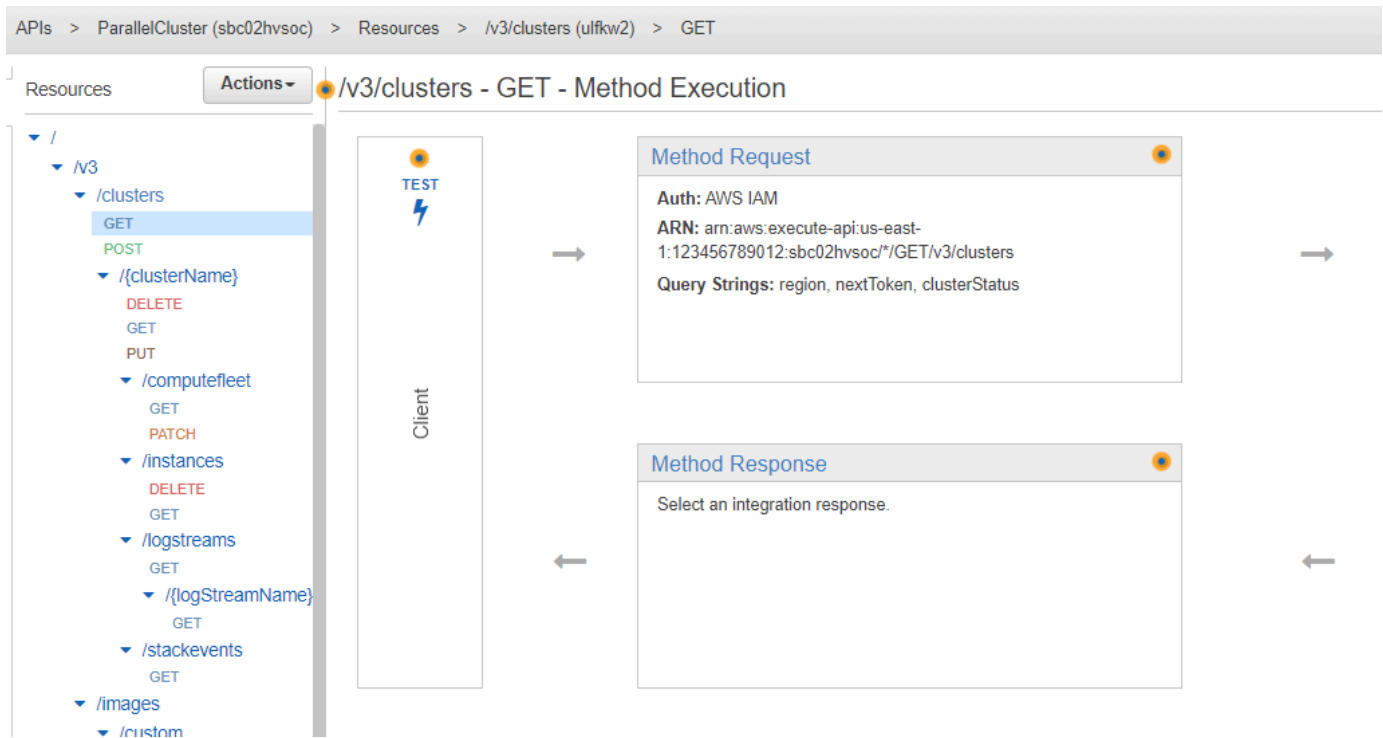




#### 4. Scegli Stages e seleziona una fase.

The screenshot displays the AWS API Gateway console interface. The breadcrumb navigation at the top reads: **APIs** > **ParallelCluster (sbc02hvsoc)** > **Stages** > **prod**. On the right side of the breadcrumb, there are links for **Show all hints** and a help icon. The left-hand navigation pane lists various API Gateway features: **APIs**, **Custom Domain Names**, **VPC Links**, **API: ParallelCluster** (highlighted), **Resources**, **Stages** (highlighted), **Authorizers**, **Gateway Responses**, **Models**, **Resource Policy**, **Documentation**, **Dashboard**, **Settings**, **Usage Plans**, **API Keys**, **Client Certificates**, and **Settings**. The main content area is titled **prod Stage Editor** and includes **Delete Stage** and **Configure Tags** buttons. A blue banner at the top of the editor displays the **Invoke URL**: `https://sbc02hvsoc.execute-api.us-east-1.amazonaws.com/prod`. Below this, there are tabs for **Settings**, **Logs/Tracing**, **Stage Variables**, **SDK Generation**, and **Export**. Under the **Settings** tab, there are sub-tabs for **Deployment History**, **Documentation History**, and **Canary**. The **Cache Settings** section includes an **Enable API cache** checkbox. The **Default Method Throttling** section provides instructions and shows **Rate** set to 100 requests per second and **Burst** set to 10 requests, with the **Enable throttling** checkbox checked. The **Web Application Firewall (WAF)** section includes a **Web ACL** dropdown menu set to **None** and a **Create Web ACL** link. The **Client Certificate** section includes a **Certificate** dropdown menu set to **None**.

5. Nota l'URL fornito da API Gateway per accedere o richiamare la tua API. È evidenziato in blu.
6. Scegli Risorse e seleziona **GET** sotto **/clusters**.
7. Scegli l'icona TEST, quindi scorri verso il basso e scegli l'icona TEST.



/clusters GETViene visualizzata la risposta al tuo messaggio.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfwk2) > GET

Show all hints ?

Resources Actions

Method Execution /v3/clusters - GET - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

**Path**

No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.

**Request:** /v3/clusters

**Status:** 200

**Latency:** 3203 ms

**Response Body**

**Query Strings**

{clusters}

param1=value1&param2=value2

**Headers**

{clusters}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

**Stage Variables**

No stage variables exist for this method.

**Client Certificate**

No client certificates have been generated.

```
{
 "clusters": [
 {
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/test-cluster/4450d850-b684-11ec-84a7-0a047567c9f3",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "clusterName": "test-cluster",
 "clusterStatus": "CREATE_COMPLETE",
 "region": "us-east-1",
 "version": "3.1.2"
 }
]
}
```

**Response Headers**

```
{"Content-Length": "360", "X-Amzn-Trace-Id": "Root=1-62686455-c1cf243417b2721e33822ac5;Sampled=1", "Content-Type": "application/json"}
```

Logs

### Fase 3: Preparare e testare un client di esempio per richiamare l'API

Clona il codice AWS ParallelCluster sorgente `api` nella directory e installa le librerie client Python. `cd`

- ```
$ git clone -b v${PCLUSTER_VERSION} https://github.com/aws/aws-parallelcluster aws-parallelcluster-v${PCLUSTER_VERSION}
cd aws-parallelcluster-v${PCLUSTER_VERSION}/api
```

```
$ pip3 install client/src
```

- Torna alla tua directory utente principale.

- Esporta l'URL di base dell'API Gateway utilizzato dal client durante l'esecuzione.

```
$ export PCLUSTER_API_URL=$( aws cloudformation describe-stacks
--stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiInvokeUrl`.OutputValue' --output text )
echo "export PCLUSTER_API_URL=${PCLUSTER_API_URL}" |tee -a ~/.bashrc
```

- Esporta il nome di un cluster utilizzato dal client per creare un cluster.

```
$ export CLUSTER_NAME="test-api-cluster"
echo "export CLUSTER_NAME=${CLUSTER_NAME}" |tee -a ~/.bashrc
```

- Esegui i seguenti comandi per memorizzare le credenziali utilizzate dal client di esempio per accedere all'API.

```
$ export PCLUSTER_API_USER_ROLE=$( aws cloudformation describe-
stacks --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiUserRole`.OutputValue' --output text )
echo "export PCLUSTER_API_USER_ROLE=${PCLUSTER_API_USER_ROLE}" |tee -a ~/.bashrc
```

Passaggio 4: Copia lo script del codice del client ed esegui i test del cluster

- Copia il seguente codice client di esempio `test_pcluster_client.py` nella tua directory utente principale. Il codice client richiede le seguenti operazioni:
 - Crea il cluster .
 - Descrivi il cluster.
 - Elenca i cluster.
 - Descrivi la flotta di elaborazione.
 - Descrivi le istanze del cluster.

```
# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
this
# software and associated documentation files (the "Software"), to deal in the
Software
```

```

# without restriction, including without limitation the rights to use, copy,
# modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and
# to
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
# COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)
    fleet_ops = cluster_compute_fleet_api.ClusterComputeFleetApi(api_client)
    instance_ops = cluster_instances_api.ClusterInstancesApi(api_client)

```

```
# Create cluster
build_done = False
try:
    with open('cluster-config.yaml', encoding="utf-8") as f:
        body = CreateClusterRequestContent(cluster_name=cluster_name,
cluster_configuration=f.read())
        api_response = cluster_ops.create_cluster(body, region=region)
except pcluster_client.ApiException as e:
    print("Exception when calling create_cluster: %s\n" % e)
    build_done = True
time.sleep(60)

# Confirm cluster status with describe_cluster
while not build_done:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('CREATE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
        elif api_response.cluster_status == ClusterStatus('CREATE_COMPLETE'):
            print('READY!')
            build_done = True
        else:
            print('ERROR!!!!')
            build_done = True
    except pcluster_client.ApiException as e:
        print("Exception when calling describe_cluster: %s\n" % e)

# List clusters
try:
    api_response = cluster_ops.list_clusters(region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling list_clusters: %s\n" % e)

# DescribeComputeFleet
try:
    api_response = fleet_ops.describe_compute_fleet(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling compute fleet: %s\n" % e)
```

```
# DescribeClusterInstances
try:
    api_response = instance_ops.describe_cluster_instances(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling describe_cluster_instances: %s\n" % e)
```

2. Crea una configurazione del cluster.

```
$ pcluster configure --config cluster-config.yaml
```

3. La libreria API Client rileva automaticamente i dettagli di configurazione dalle variabili di ambiente (ad esempio `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, o `AWS_SESSION_TOKEN`) o `$HOME/.aws`. Il comando seguente imposta il ruolo IAM corrente a quello designato. `ParallelClusterApiUserRole`

```
$ eval $(aws sts assume-role --role-arn ${PCLUSTER_API_USER_ROLE} --role-session-name ApiTestSession | jq -r '.Credentials | "export AWS_ACCESS_KEY_ID=\(.AccessKeyId)\nexport AWS_SECRET_ACCESS_KEY=\(.SecretAccessKey)\nexport AWS_SESSION_TOKEN=\(.SessionToken)\n"')
```

Errore da tenere d'occhio:

Se visualizzi un errore simile al seguente, significa che hai già dato per scontato che il `ParallelClusterApiUserRole` e il tuo sia `AWS_SESSION_TOKEN` scaduto.

```
An error occurred (AccessDenied) when calling the AssumeRole operation:
User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX/ApiTestSession
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::XXXXXXXXXXXX:role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX
```

Eliminare il ruolo e quindi eseguire nuovamente il `aws sts assume-role` comando per utilizzare. `ParallelClusterApiUserRole`

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
```

```
unset AWS_ACCESS_KEY_ID
```

Per fornire le attuali autorizzazioni utente per l'accesso all'API, è necessario [espandere la Resource Policy](#).

4. Esegui il comando seguente per testare il client di esempio.

```
$ python3 test_pcluster_client.py
{'cluster_configuration': 'Region: us-east-1\n'
                          'Image:\n'
                          '  Os: alinux2\n'
                          'HeadNode:\n'
                          '  InstanceType: t2.micro\n'
                          '  Networking . . . :\n'
                          '    SubnetId: subnet-1234567890abcdef0\n'
                          '  Ssh:\n'
                          '    KeyName: adpc\n'
                          'Scheduling:\n'
                          '  Scheduler: slurm\n'
                          '  SlurmQueues:\n'
                          '    - Name: queue1\n'
                          '      ComputeResources:\n'
                          '        - Name: t2micro\n'
                          '          InstanceType: t2.micro\n'
                          '          MinCount: 0\n'
                          '          MaxCount: 10\n'
                          '      Networking . . . :\n'
                          '        SubnetIds:\n'
                          '          - subnet-1234567890abcdef0\n',
  'cluster_name': 'test-api-cluster'}
{'cloud_formation_stack_status': 'CREATE_IN_PROGRESS',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
  'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-not-delete...'},
  'cluster_name': 'test-api-cluster',
  'cluster_status': 'CREATE_IN_PROGRESS',
  'compute_fleet_status': 'UNKNOWN',
  'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'region': 'us-east-1',
```



```

'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'
.
.
.
. . . working . . . {'cloud_formation_stack_status': 'CREATE_COMPLETE',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'CREATE_COMPLETE',
'compute_fleet_status': 'RUNNING',
'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 21, 46,
tzinfo=tzlocal()),
'private_ip_address': '172.31.27.153',
'public_ip_address': '52.90.156.51',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'
READY!

```

Passaggio 5: Copia lo script del codice del client ed elimina il cluster

1. Copia il seguente codice client di esempio `indelete_cluster_client.py`. Il codice client richiede l'eliminazione del cluster.

```

# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
# this
# software and associated documentation files (the "Software"), to deal in the
# Software

```

```
# without restriction, including without limitation the rights to use, copy,
# modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and
# to
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
# COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)

    # Delete the cluster
    gone = False
```

```

try:
    api_response = cluster_ops.delete_cluster(cluster_name, region=region)
except pcluster_client.ApiException as e:
    print("Exception when calling delete_cluster: %s\n" % e)
time.sleep(60)

# Confirm cluster status with describe_cluster
while not gone:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('DELETE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
    except pcluster_client.ApiException as e:
        gone = True
        print("DELETE COMPLETE or Exception when calling describe_cluster: %s
\n" % e)

```

2. Esegui il comando seguente per eliminare il cluster.

```

$ python3 delete_cluster_client.py
{'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 53, 48,
tzinfo=tzlocal()),
'private_ip_address': '172.31.17.132',
'public_ip_address': '34.201.100.37',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],

```

```
'version': '3.1.3'}
.
.
.
. . . working . . . {'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
. . . working . . . DELETE COMPLETE or Exception when calling describe_cluster:
(404)
Reason: Not Found
.
.
.
HTTP response body: {"message": "Cluster 'test-api-cluster' does not exist or
belongs to an incompatible ParallelCluster major version."}
```

3. Al termine del test, annullate l'impostazione delle variabili di ambiente.

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

Fase 6: pulizia

Puoi usare Console di gestione AWS o AWS CLI per eliminare la tua API.

1. Dalla CloudFormation console, scegli lo stack API, quindi scegli Elimina.
2. Esegui il comando seguente se usi il AWS CLI.

Usando CloudFormation.

```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```

Creazione di un cluster con Slurm contabilità

Scopri come configurare e creare un cluster con Slurm contabilità. Per ulteriori informazioni, consulta [Slurmcontabilità con AWS ParallelCluster](#).

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

In questo tutorial, si utilizza un [modello di CloudFormation creazione rapida \(us-east-1\) per creare un database serverless per MySQL](#). [Amazon Aurora](#) Il modello indica di CloudFormation creare tutti i componenti necessari per distribuire un database Amazon Aurora serverless sullo stesso VPC del cluster. Il modello crea anche una configurazione di rete e sicurezza di base per la connessione tra il cluster e il database.

Note

A partire dalla versione 3.3.0, supporta AWS ParallelCluster Slurm contabilità con il parametro di configurazione del cluster [SlurmSettings/Database](#).

Note

Il modello di creazione rapida funge da esempio. Questo modello non copre tutti i possibili casi d'uso per un Slurm server di database contabile. È tua responsabilità creare un server di database con la configurazione e la capacità appropriate per i tuoi carichi di lavoro di produzione.

Prerequisiti:

- AWS ParallelCluster [è installato](#).
- AWS CLI [è installato e configurato](#).
- Hai una [coppia di EC2 chiavi Amazon](#).

- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.
- La regione in cui distribuisce il modello di creazione rapida supporta Amazon Aurora MySQL serverless v2. Per ulteriori informazioni, consulta [Aurora Serverless v2 with Aurora](#) MySQL.

Fase 1: Creare il VPC e le sottoreti per AWS ParallelCluster

Per utilizzare il modello fornito CloudFormation per Slurm database di contabilità, è necessario disporre del VPC per il cluster pronto. È possibile eseguire questa operazione manualmente o come parte della [Configura e crea un cluster con AWS ParallelCluster interfaccia a riga di comando](#) procedura. Se lo hai già utilizzato AWS ParallelCluster, potresti avere un VPC pronto per la distribuzione del cluster e del server di database.

Fase 2: Creare lo stack di database

Usa il [modello CloudFormation quick-create \(us-east-1\) per creare](#) uno stack di database per Slurm contabilità. Il modello richiede i seguenti input:

- Credenziali del server di database, in particolare il nome utente e la password dell'amministratore.
- Dimensionamento del cluster Amazon Aurora serverless. Dipende dal caricamento previsto del cluster.
- Parametri di rete, in particolare il VPC e le sottoreti di destinazione o i blocchi CIDR per la creazione delle sottoreti.

Seleziona le credenziali e le dimensioni appropriate per il tuo server di database. Per le opzioni di rete, è necessario utilizzare lo stesso VPC su cui è distribuito il AWS ParallelCluster cluster. È possibile creare le sottoreti per il database e passarle come input al modello. Oppure, fornisci due blocchi CIDR disgiunti per le due sottoreti e lascia che il CloudFormation modello crei le due sottoreti per i blocchi CIDR. Assicurati che i blocchi CIDR non si sovrappongano alle sottoreti esistenti. Se i blocchi CIDR si sovrappongono alle sottoreti esistenti, lo stack non viene creato.

La creazione del server del database richiede diversi minuti.

Fase 3: Creare un cluster con Slurm contabilità abilitata

Il CloudFormation modello fornito genera uno CloudFormation stack con alcuni output definiti. Da Console di gestione AWS, è possibile visualizzare gli output nella scheda Output nella visualizzazione

in pila. CloudFormation Per abilitare il Slurm accounting, alcuni di questi output devono essere utilizzati nel file di configurazione del AWS ParallelCluster cluster:

- DatabaseHost: Utilizzato per il parametro di configurazione del [Uri](#)cluster [SlurmSettingsDatabase](#)//.
- DatabaseAdminUser: utilizzato per il valore del parametro di configurazione [UserName](#)del cluster [SlurmSettingsDatabase](#)//.
- DatabaseSecretArn: utilizzato per il parametro di configurazione [PasswordSecretArn](#)del cluster [SlurmSettingsDatabase](#)//.
- DatabaseClientSecurityGroup: Questo è il gruppo di sicurezza collegato al nodo principale del cluster definito nel parametro di [SecurityGroups](#)configurazione [HeadNodeNetworking](#)//.

Aggiorna i Database parametri del file di configurazione del cluster con i valori di output. Usa la [pcluster](#) CLI per creare il cluster.

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

Dopo aver creato il cluster, puoi iniziare a utilizzare Slurm comandi di contabilità come `sacctmgr osacct`.

Creazione di un cluster con un cluster esterno Slurmdbd contabilità

Scopri come configurare e creare un cluster con elementi esterni Slurmdbd contabilità. Per ulteriori informazioni, consulta [Slurm contabilità con AWS ParallelCluster](#).

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS i servizi utilizzati](#) da. AWS ParallelCluster

L' AWS ParallelCluster interfaccia utente è basata su un'architettura serverless ed è possibile utilizzarla all'interno di AWS Free Tier categoria per la maggior parte dei casi. Per ulteriori informazioni, consulta [AWS ParallelCluster Costi dell'interfaccia utente](#).

In questo tutorial, utilizzi un modello di AWS CloudFormation creazione rapida per creare i componenti necessari per distribuire un'istanza Slurmdbd sullo stesso VPC del cluster. Il modello crea una configurazione di rete e sicurezza di base per la connessione tra il cluster e il database.

 Note

A partire da `version 3.10.0`, AWS ParallelCluster supporta Slurmdbd esterno con il parametro di configurazione del cluster. `SlurmSettings / ExternalSlurmdbd`

 Note

Il modello di creazione rapida funge da esempio. Questo modello non copre tutti i possibili casi d'uso. È tua responsabilità creare uno Slurmdbd esterno con la configurazione e la capacità appropriate per i tuoi carichi di lavoro di produzione.

Prerequisiti:

- AWS ParallelCluster [è](#) installato.
- AWS CLI [è installato e configurato](#).
- Hai una [coppia di chiavi Amazon Elastic Compute Cloud](#).
- Hai un AWS Identity and Access Management ruolo con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.
- Hai un Slurm database contabile. Per seguire un tutorial sulla creazione Slurm database di contabilità, segui i passaggi 1 e 2 in [Creare lo stack del database di contabilità Slurm](#).

Passaggio 1: creare lo stack Slurmdbd

In questo tutorial, usa un [template di CloudFormation creazione rapida \(us-east-1\) per creare](#) uno stack Slurmdbd. Il modello richiede i seguenti input:

Rete

- VPCId: L'ID VPC per avviare l'istanza Slurmdbd.
- SubnetId: L'ID di sottorete per avviare l'istanza Slurmdbd.
- PrivatePrefix: il prefisso CIDR del VPC.
- PrivateIp: un IP privato secondario da assegnare all'istanza Slurmdbd.

Connessioni database

- **DBMSClientSG**: Il gruppo di sicurezza da collegare all'istanza Slurmdbd. Questo gruppo di sicurezza dovrebbe consentire le connessioni tra il server del database e l'istanza Slurmdbd.
- **DBMSDatabaseNome**: il nome del database.
- **DBMSUsername**: il nome utente del database.
- **DBMSPasswordSecretArn**: Il segreto contenente la password per il database.
- **DBMSUri**: l'URI del server del database.

Impostazioni a livello di istanza

- **InstanceType**: un tipo di istanza da utilizzare per l'istanza slurmdbd.
- **KeyName**: una coppia di EC2 chiavi Amazon da usare per l'istanza slurmdbd.

Impostazioni Slurmdbd

- **AMIID**: un AMI dell'istanza Slurmdbd. L'AMI dovrebbe essere un ParallelCluster AMI. La versione dell' ParallelCluster AMI determina la versione di Slurmdbd.
- **MungeKeySecretArn**: Il segreto contenente la chiave munge da utilizzare per autenticare le comunicazioni tra Slurmdbd e i cluster.
- **SlurmdbdPort**: Un numero di porta utilizzato da slurmdbd.
- **EnableSlurmdbdSystemService**: abilita slurmdbd come servizio di sistema e lo fa funzionare all'avvio di un'istanza.

Warning

Se il database è stato creato da una versione diversa di SlurmDB, non utilizzare Slurmdbd come servizio di sistema.

Se il database contiene un numero elevato di voci, Slurm Database Daemon (SlurmDBD) potrebbe richiedere decine di minuti per aggiornare il database e non rispondere durante questo intervallo di tempo.

Prima dell'aggiornamento SlurmDB, effettua un backup del database. Per ulteriori informazioni, consultare la [.Slurm documentazione](#).

Fase 2: Creare un cluster con elementi esterni Slurmdbd abilitato

Il CloudFormation modello fornito genera uno CloudFormation stack con alcuni output definiti.

Da Console di gestione AWS, visualizza la scheda Output nella CloudFormation pila per esaminare le entità create. Per abilitare il Slurm contabilizzazione, alcuni di questi output devono essere utilizzati nel file di AWS ParallelCluster configurazione:

- SlurmdbdPrivatelP: [Utilizzato per il parametro di configurazione del cluster SlurmSettings/ExternalSlurmdbd/Host.](#)
- SlurmdbdPort: Utilizzato per il valore del [SlurmSettings](#) parametro di configurazione del cluster [ExternalSlurmdbd//Port.](#)
- AccountingClientSecurityGroup: [Questo è il gruppo di sicurezza collegato al nodo principale del cluster definito nel parametro di configurazione HeadNode/Networking /. AdditionalSecurityGroups](#)

Inoltre, dalla scheda Parametri nella vista stack: CloudFormation

- MungeKeySecretArn: utilizzato per il valore del [SlurmSettings](#) parametro di configurazione [/MungeKeySecretArn](#) cluster.

Aggiorna i parametri del database dei file dei file di configurazione del cluster con i valori di output. Usa pcluster per creare il cluster AWS CLI .

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

Dopo aver creato il cluster, puoi iniziare a usare Slurm comandi di contabilità come `sacctmgr` o `sacct`.

Warning

Traffico tra ParallelCluster e verso l'esterno SlurmDB non è crittografato. Si consiglia di eseguire il cluster e l'esterno SlurmDB in una rete affidabile.

Ripristino a una versione precedente del documento AWS Systems Manager

Scoprite come ripristinare una versione precedente del documento AWS Systems Manager. Per ulteriori informazioni sui documenti SSM, vedere [AWS Systems Manager Documents](#) nella AWS Systems Manager User Guide.

Quando si utilizza l'interfaccia a riga di AWS ParallelCluster comando (CLI) o l'API, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Prerequisiti:

- E Account AWS con le autorizzazioni per gestire i documenti SSM.
- AWS CLI [È installato e configurato](#).

Ripristina una versione precedente del documento SSM

1. Nel tuo terminale, esegui il seguente comando per ottenere l'elenco dei documenti SSM esistenti di cui sei proprietario.

```
$ aws ssm list-documents --document-filter "key=Owner,value=Self"
```

2. Ripristina una versione precedente di un documento SSM. In questo esempio, torniamo a una versione precedente del documento. `SessionManagerRunShell` È possibile utilizzare il `SessionManagerRunShell` documento SSM per personalizzare ogni sessione della shell SSM che si avvia.
 - a. Trovate il `DocumentVersion` parametro per `SessionManagerRunShell` eseguendo il seguente comando:

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
```

```

    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {
        "Name": "linuxcmd",
        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then
source /opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/
bash"
      }
    ],
    "PlatformTypes": [
      "Windows",
      "Linux",
      "MacOS"
    ],
    "DocumentType": "Session",
    "SchemaVersion": "1.0",
    "LatestVersion": "2",
    "DefaultVersion": "1",
    "DocumentFormat": "JSON",
    "Tags": []
  }
}

```

La versione più recente è 2.

- b. Tornate alla versione precedente eseguendo il comando seguente:

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-version 2
```

3. Verificate che la versione del documento sia stata ripristinata eseguendo nuovamente il describe-document comando:

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",

```

```
"CreateDate": "2023-02-20T19:04:32.390000+00:00",
"Status": "Active",
"DocumentVersion": "1",
"Parameters": [
  {
    "Name": "linuxcmd",
    "Type": "String",
    "Description": "The command to run on connection...",
    "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then source /
opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/bash"
  }
],
"PlatformTypes": [
  "Windows",
  "Linux",
  "MacOS"
],
"DocumentType": "Session",
"SchemaVersion": "1.0",
"LatestVersion": "1",
"DefaultVersion": "1",
"DocumentFormat": "JSON",
"Tags": []
}
```

La versione più recente è 1.

Creazione di un cluster con CloudFormation

Scopri come creare un cluster con una risorsa AWS ParallelCluster CloudFormation personalizzata. Per ulteriori informazioni, consulta [AWS CloudFormation risorsa personalizzata](#).

Quando si utilizza AWS ParallelCluster, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Prerequisiti:

- AWS CLI [È installato e configurato](#).
- Una [coppia di chiavi Amazon EC2](#).

- Un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

Creazione di cluster con uno CloudFormation stack di creazione rapida

In questo tutorial, si utilizza uno stack di creazione rapida per distribuire un CloudFormation modello che crea un cluster e le seguenti risorse: AWS

- Un root CloudFormation stack creato utilizzando uno stack di creazione rapida. CloudFormation
- CloudFormation Stack annidati che includono policy predefinite, configurazione VPC predefinita e un provider di risorse personalizzato.
- Un esempio di stack di AWS ParallelCluster cluster e un cluster a cui è possibile accedere ed eseguire lavori.

Crea un cluster con AWS CloudFormation

1. Accedi alla Console di gestione AWS.
2. Apri il [link di CloudFormation creazione rapida](#) per creare le seguenti risorse nella CloudFormation console:
 - Uno CloudFormation stack annidato con un VPC con una sottorete pubblica e privata per l'esecuzione rispettivamente del nodo principale del cluster e dei nodi di calcolo.
 - Uno CloudFormation stack annidato con una risorsa personalizzata per la gestione del cluster AWS ParallelCluster .
 - Uno CloudFormation stack annidato con le politiche predefinite per la gestione del cluster.
 - Uno stack principale per CloudFormation gli stack annidati.
 - Un AWS ParallelCluster cluster con lo Slurm scheduler e un numero definito di nodi di calcolo.

CloudFormation > Stacks > Create stack

Quick create stack

Template

Template URL

https://pcluster-cfn-us-east-2.s3.amazonaws.com/parallelcluster/3.5.0/templates/custom_resource/cluster-1-click.yaml

Stack description

AWS ParallelCluster CloudFormation Cluster

Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AvailabilityZone

Availability zone where instances will be launched

KeyName

KeyPair to login to the head node

Capabilities

**The following resource(s) require capabilities: [AWS::CloudFormation::Stack]**

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: {0}. Check the capabilities of these resources. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.☒ I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND

Cancel

Create change set

Create stack

3. Nella sezione Parametri dello stack di creazione rapida, immettete i valori per i seguenti parametri:
 - a. Per KeyName, inserisci il nome della tua coppia di chiavi Amazon EC2.

- b. Per AvailabilityZone, scegli una AZ per i nodi del cluster, us-east-1a ad esempio.
4. Seleziona le caselle per confermare ciascuna delle funzionalità di accesso nella parte inferiore della pagina.
5. Seleziona Crea stack.
6. Attendi che lo CloudFormation stack raggiunga lo CREATE_COMPLETE stato.

Creazione di cluster con l'interfaccia CloudFormation a riga di comando (CLI)

In questo tutorial, si utilizza l'interfaccia a riga di AWS comando (CLI) CloudFormation per distribuire un CloudFormation modello che crea un cluster.

Crea le seguenti risorse: AWS

- Una pila principale creata utilizzando uno CloudFormation stack di creazione CloudFormation rapida.
- CloudFormation Stack annidati che includono policy predefinite, configurazione VPC predefinita e un provider di risorse personalizzato.
- Uno stack di AWS ParallelCluster cluster di esempio e un cluster a cui è possibile accedere ed eseguire lavori.

Sostituiscilo *inputs highlighted in red*, ad esempio *keypair*, con i tuoi valori.

Crea un cluster con AWS CloudFormation

1. Crea un CloudFormation modello denominato `cluster_template.yaml` con il seguente contenuto:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
  KeyName:
    Description: KeyPair to login to the head node
    Type: AWS::EC2::KeyPair::KeyName

  AvailabilityZone:
    Description: Availability zone where instances will be launched
```



```

    Type: AWS::EC2::AvailabilityZone::Name
    Default: us-east-2a

Mappings:
  ParallelCluster:
    Constants:
      Version: 3.15.0

Resources:
  PclusterClusterProvider:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
        - { Version: !FindInMap [ParallelCluster, Constants, Version] }

  PclusterVpc:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        PublicCIDR: 10.0.0.0/24
        PrivateCIDR: 10.0.16.0/20
        AvailabilityZone: !Ref AvailabilityZone
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/networking/public-private-
          ${Version}.cfn.json
        - { Version: !FindInMap [ParallelCluster, Constants, Version] }

  PclusterCluster:
    Type: Custom::PclusterCluster
    Properties:
      ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
      ClusterName: !Sub 'c-${AWS::StackName}'
      ClusterConfiguration:
        Image:
          Os: alinux2
        HeadNode:
          InstanceType: t2.medium
          Networking:
            SubnetId: !GetAtt [ PclusterVpc , Outputs.PublicSubnetId ]
        Ssh:
          KeyName: !Ref KeyName

```

```

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
      ComputeResources:
        - Name: queue0-cr0
          InstanceType: t2.micro
      Networking:
        SubnetIds:
          - !GetAtt [ PclusterVpc , Outputs.PrivateSubnetId ]
Outputs:
  HeadNodeIp:
    Description: The Public IP address of the HeadNode
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]

```

2. Esegui il seguente comando AWS CLI per distribuire lo CloudFormation stack per la creazione e la gestione del cluster.

```

$ aws cloudformation deploy --template-file ./cluster_template.yaml \
  --stack-name mycluster \
  --parameter-overrides KeyName=keypair \
    AvailabilityZone=us-east-2b \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND

```

Visualizza l'output del cluster CloudFormation

Visualizza l'output del CloudFormation cluster per ottenere dettagli utili sul cluster. La `ValidationMessages` proprietà aggiunta fornisce l'accesso ai messaggi di convalida dalle operazioni di creazione e aggiornamento del cluster.

1. Vai alla [CloudFormation console](#) e seleziona lo stack che include la tua risorsa AWS ParallelCluster personalizzata.
2. Scegli i dettagli dello stack e seleziona la scheda Output.

Outputs (2)			
Search outputs			
Key	Value	Description	
HeadNodeIp	1.2.3.4	The Public IP address of the HeadNode	
ValidationMessages	[[{"level": "WARNING", "type": "KeyPairValidator", "message": "If you do not specify a key pair, you can't connect to the instance unless you choose an AMI that is configured to allow users another way to log in"}]]	Any warnings from cluster create or update operations.	

I messaggi di convalida potrebbero essere troncati. Per ulteriori informazioni su come recuperare i log, vedere. [AWS ParallelCluster risoluzione dei problemi](#)

Accedi al tuo cluster

Accedi al cluster.

ssh nel nodo principale del cluster

- Una volta completata la distribuzione CloudFormation dello stack, ottieni l'indirizzo IP del nodo principale con il seguente comando:

```
$ HEAD_NODE_IP=$(aws cloudformation describe-stacks --stack-name=mycluster --query "Stacks|[0].Outputs[?OutputKey=='HeadNodeIp']|[0].OutputValue" --output=text)
```

È inoltre possibile recuperare l'indirizzo IP del nodo principale dal HeadNodeIp parametro nella scheda Outputs dello stack del cluster nella console. CloudFormation

Puoi trovare l'indirizzo IP del nodo principale qui perché è stato aggiunto nella Outputs sezione del CloudFormation modello di cluster, in particolare per questo cluster di esempio.

- Connect al nodo principale del cluster eseguendo il comando seguente:

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

Eliminazione

Elimina il cluster.

1. Esegui il seguente comando AWS CLI per eliminare lo CloudFormation stack e il cluster.

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. Controlla lo stato di eliminazione dello stack eseguendo il comando seguente.

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

Implementa l' ParallelCluster API con Terraform

In questo tutorial, definirai un semplice progetto Terraform per implementare un'API. ParallelCluster

Prerequisiti

- Terraform v1.5.7+ è installato.
- Ruolo IAM con le autorizzazioni per distribuire l'API. ParallelCluster Per informazioni, consulta [the section called "Autorizzazioni richieste"](#).

Definisci un progetto Terraform

In questo tutorial, definirai un progetto Terraform.

1. Crea una directory chiamata `my-pcluster-api`.

Tutti i file che creerai si troveranno all'interno di questa directory.

2. Crea il file `provider.tf` per configurare il AWS provider.

```
provider "aws" {  
  region = var.region  
  profile = var.profile  
}
```

3. Crea il file `main.tf` per definire le risorse utilizzando il ParallelCluster modulo.

```
module "parallelcluster_pcluster_api" {  
  source = "aws-tf/parallelcluster/aws//modules/pcluster_api"  
  version = "1.1.0"  
  
  region = var.region
```

```
api_stack_name      = var.api_stack_name
api_version         = var.api_version

parameters = {
    EnableIamAdminAccess = "true"
}
}
```

4. Crea il file `variables.tf` per definire le variabili che possono essere iniettate per questo progetto.

```
variable "region" {
    description = "The region the ParallelCluster API is deployed in."
    type       = string
    default    = "us-east-1"
}

variable "profile" {
    type       = string
    description = "The AWS profile used to deploy the clusters."
    default    = null
}

variable "api_stack_name" {
    type       = string
    description = "The name of the CloudFormation stack used to deploy the ParallelCluster API."
    default    = "ParallelCluster"
}

variable "api_version" {
    type       = string
    description = "The version of the ParallelCluster API."
}
```

5. Crea il file `terraform.tfvars` per impostare valori arbitrari per le variabili.

Il file seguente implementa un' `ParallelCluster API 3.11.1 us-east-1` utilizzando il nome dello stack. `MyParallelClusterAPI-3111` Potrai fare riferimento a questa distribuzione dell' `ParallelCluster API` utilizzando il nome dello stack.

 Note

L'`api_version` assegnazione nel codice seguente può essere sostituita con qualsiasi versione supportata AWS ParallelCluster .

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-3111"
api_version = "3.11.1"
```

6. Crea il file `outputs.tf` per definire gli output restituiti da questo progetto.

```
output "pcluster_api_stack_outputs" {
  value = module.parallelcluster_pcluster_api.stack_outputs
}
```

La cartella del progetto è:

```
my-pcluster-api
### main.tf - Terraform entrypoint to define the resources using the
ParallelCluster module.
### outputs.tf - Defines the outputs returned by Terraform.
### providers.tf - Configures the AWS provider.
### terraform.tfvars - Set the arbitrary values for the variables, i.e. region,
PCAPI version, PCAPI stack name
### variables.tf - Defines the variables, e.g. region, PCAPI version, PCAPI stack
name.
```

Distribuzione dell'API

Per distribuire l'API, esegui i comandi Terraform standard nell'ordine.

1. Compila il progetto:

```
terraform init
```

2. Definisci il piano di implementazione:

```
terraform plan -out tfplan
```

3. Implementa il piano:

```
terraform apply tfplan
```

Autorizzazioni richieste

Sono necessarie le seguenti autorizzazioni per distribuire l'API con Terraform: ParallelCluster

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:GetTemplate"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormationRead"
    },
    {
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/MyParallelClusterAPI*",
      "Effect": "Allow",
      "Sid": "CloudFormationWrite"
    },
    {
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": [
```

```

        "arn:aws:cloudformation:us-east-1:111122223333:aws:transform/
Include",
        "arn:aws:cloudformation:us-east-1:111122223333:aws:transform/
Serverless-2016-10-31"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormationTransformWrite"
},
{
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:us-east-1:111122223333:*-aws-parallelcluster/
parallelcluster/*/api/ParallelCluster.openapi.yaml",
        "arn:aws:s3:us-east-1:111122223333:*-aws-parallelcluster/
parallelcluster/*/layers/aws-parallelcluster/lambda-layer.zip"
    ],
    "Effect": "Allow",
    "Sid": "S3ParallelClusterArtifacts"
},
{
    "Action": [
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:GetRole",
        "iam:CreatePolicy",
        "iam:DeletePolicy",
        "iam:GetPolicy",
        "iam:GetRolePolicy",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam:DeleteRolePolicy",
        "iam:ListPolicyVersions"
    ],
    "Resource": [
        "arn:aws:iam::111122223333:role/*",
        "arn:aws:iam::111122223333:policy/*"
    ],
    "Effect": "Allow",
    "Sid": "IAM"
},
{

```



```

        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:role/ParallelClusterLambdaRole-*",
            "arn:aws:iam::111122223333:role/APIGatewayExecutionRole-*"
        ],
        "Effect": "Allow",
        "Sid": "IAMPassRole"
    },
    {
        "Action": [
            "lambda:CreateFunction",
            "lambda:DeleteFunction",
            "lambda:GetFunction",
            "lambda:PublishLayerVersion",
            "lambda:DeleteLayerVersion",
            "lambda:GetLayerVersion",
            "lambda:TagResource",
            "lambda:UntagResource"
        ],
        "Resource": [
            "arn:aws:lambda:us-east-1:111122223333:layer:PCLayer-*",
            "arn:aws:lambda:us-east-1:111122223333:function:*-
ParallelClusterFunction-*"
        ],
        "Effect": "Allow",
        "Sid": "Lambda"
    },
    {
        "Action": [
            "logs:CreateLogGroup",
            "logs>DeleteLogGroup",
            "logs:DescribeLogGroups",
            "logs:PutRetentionPolicy",
            "logs:TagLogGroup",
            "logs:UntagLogGroup"
        ],
        "Resource": [
            "arn:aws:logs:us-east-1:111122223333:log-group:/aws/lambda/*-
ParallelClusterFunction-*"
        ],
        "Effect": "Allow",
        "Sid": "Logs"
    }

```

```
    },
    {
      "Action": [
        "apigateway:DELETE",
        "apigateway:GET",
        "apigateway:PATCH",
        "apigateway:POST",
        "apigateway:PUT",
        "apigateway:UpdateRestApiPolicy"
      ],
      "Resource": [
        "arn:aws:apigateway:us-east-1::/restapis",
        "arn:aws:apigateway:us-east-1::/restapis/*",
        "arn:aws:apigateway:us-east-1::/tags/*"
      ],
      "Effect": "Allow",
      "Sid": "APIGateway"
    }
  ]
}
```

Creare un cluster con Terraform

Quando si utilizza AWS ParallelCluster, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [the section called “AWS servizi usati da AWS ParallelCluster”](#).

Prerequisiti

- Terraform v1.5.7+ è installato.
- [the section called “AWS ParallelCluster API”](#) v3.8.0+ è distribuito nel tuo account. Per informazioni, consulta [the section called “Implementa l' ParallelCluster API con Terraform”](#).
- Ruolo IAM con le autorizzazioni per richiamare l'API. ParallelCluster Vedi [Autorizzazioni richieste]

Definisci un progetto Terraform

In questo tutorial, definirai un semplice progetto Terraform per distribuire un cluster.

1. Crea una directory chiamata `my-clusters`

Tutti i file che creerai si troveranno all'interno di questa directory.

2. Crea il file `terraform.tf` per importare il ParallelCluster provider.

```
terraform {
  required_version = ">= 1.5.7"
  required_providers {
    aws-parallelcluster = {
      source = "aws-tf/aws-parallelcluster"
      version = "~> 1.0"
    }
  }
}
```

3. Crea il file `providers.tf` per configurare i AWS provider ParallelCluster e.

```
provider "aws" {
  region = var.region
  profile = var.profile
}

provider "aws-parallelcluster" {
  region      = var.region
  profile      = var.profile
  api_stack_name = var.api_stack_name
  use_user_role = true
}
```

4. Crea il file `main.tf` per definire le risorse utilizzando il ParallelCluster modulo.

```
module "pcluster" {
  source = "aws-tf/parallelcluster/aws"
  version = "1.1.0"

  region      = var.region
  api_stack_name = var.api_stack_name
  api_version  = var.api_version
  deploy_pcluster_api = false

  template_vars      = local.config_vars
  cluster_configs     = local.cluster_configs
  config_path         = "config/clusters.yaml"
```

```
}
```

5. Crea il file `clusters.tf` per definire più cluster come variabili locali Terraform.

Note

È possibile definire più cluster all'interno dell'elemento. `cluster_config` Per ogni cluster, è possibile definire in modo esplicito le proprietà del cluster all'interno delle variabili locali (vedere `DemoCluster01`) o fare riferimento a un file esterno (vedere `DemoCluster02`).

Per esaminare le proprietà del cluster che è possibile impostare all'interno dell'elemento di configurazione, vedere [the section called “File di configurazione del cluster”](#).

Per esaminare le opzioni che è possibile impostare per la creazione di cluster, vedere [the section called “pcluster create-cluster”](#).

```
locals {
  cluster_configs = {
    DemoCluster01 : {
      region : local.config_vars.region
      rollbackOnFailure : false
      validationFailureLevel : "WARNING"
      suppressValidators : [
        "type:KeyPairValidator"
      ]
      configuration : {
        Region : local.config_vars.region
        Image : {
          Os : "alinux2"
        }
        HeadNode : {
          InstanceType : "t3.small"
          Networking : {
            SubnetId : local.config_vars.subnet
          }
          Iam : {
            AdditionalIamPolicies : [
              { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
            ]
          }
        }
      }
    }
  }
}
```

```

    }
    Scheduling : {
      Scheduler : "slurm"
      SlurmQueues : [{
        Name : "queue1"
        CapacityType : "ONDEMAND"
        Networking : {
          SubnetIds : [local.config_vars.subnet]
        }
      }
      Iam : {
        AdditionalIamPolicies : [
          { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
        ]
      }
      ComputeResources : [{
        Name : "compute"
        InstanceType : "t3.small"
        MinCount : "1"
        MaxCount : "4"
      }]
    }]
    SlurmSettings : {
      QueueUpdateStrategy : "TERMINATE"
    }
  }
}
DemoCluster02 : {
  configuration : "config/cluster_config.yaml"
}
}
}

```

6. Crea il file `config/clusters.yaml` per definire più cluster come configurazione YAML.

```

DemoCluster03:
  region: ${region}
  rollbackOnFailure: true
  validationFailureLevel: WARNING
  suppressValidators:
    - type:KeyPairValidator
  configuration: config/cluster_config.yaml
DemoCluster04:
  region: ${region}

```

```
rollbackOnFailure: false
configuration: config/cluster_config.yaml
```

7. Crea il file `config/cluster_config.yaml`, che è un file di ParallelCluster configurazione standard in cui è possibile inserire le variabili Terraform.

Per esaminare le proprietà del cluster che è possibile impostare all'interno dell'elemento di configurazione, vedere. [the section called “File di configurazione del cluster”](#)

```
Region: ${region}
Image:
  Os: alinux2
HeadNode:
  InstanceType: t3.small
  Networking:
    SubnetId: ${subnet}
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CapacityType: ONDEMAND
      Networking:
        SubnetIds:
          - ${subnet}
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
      ComputeResources:
        - Name: compute
          InstanceType: t3.small
          MinCount: 1
          MaxCount: 5
  SlurmSettings:
    QueueUpdateStrategy: TERMINATE
```

8. Crea il file `clusters_vars.tf` per definire le variabili che possono essere inserite nelle configurazioni del cluster.

Questo file consente di definire valori dinamici che possono essere utilizzati nelle configurazioni del cluster, come regione e sottorete.

Questo esempio recupera i valori direttamente dalle variabili di progetto, ma potrebbe essere necessario utilizzare una logica personalizzata per determinarli.

```
locals {
  config_vars = {
    subnet = var.subnet_id
    region = var.cluster_region
  }
}
```

9. Create il file `variables.tf` per definire le variabili che possono essere inserite per questo progetto.

```
variable "region" {
  description = "The region the ParallelCluster API is deployed in."
  type        = string
  default     = "us-east-1"
}

variable "cluster_region" {
  description = "The region the clusters will be deployed in."
  type        = string
  default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

variable "subnet_id" {
  type        = string
  description = "The id of the subnet to be used for the ParallelCluster instances."
}

variable "api_stack_name" {
  type        = string
  description = "The name of the CloudFormation stack used to deploy the ParallelCluster API."
  default     = "ParallelCluster"
```

```

}

variable "api_version" {
  type      = string
  description = "The version of the ParallelCluster API."
}

```

10. Crea il file `terraform.tfvars` per impostare valori arbitrari per le variabili.

Il file seguente distribuisce i cluster all'`eu-west-1` interno della sottorete `subnet-123456789`, utilizzando l' `ParallelCluster API 3.11.1` esistente, che è già distribuita con il nome dello stack `us-east-1 MyParallelClusterAPI-3111`

```

region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-3111"
api_version = "3.11.1"

cluster_region = "eu-west-1"
subnet_id = "subnet-123456789"

```

11. Crea il file per definire gli output restituiti `outputs.tf` da questo progetto.

```

output "clusters" {
  value = module.pcluster.clusters
}

```

La cartella del progetto è:

```

my-clusters
### config
#   ### cluster_config.yaml - Cluster configuration, where terraform variables can
#   be injected..
#   ### clusters.yaml - File listing all the clusters to deploy.
### clusters.tf - Clusters defined as Terraform local variables.
### clusters_vars.tf - Variables that can be injected into cluster configurations.
### main.tf - Terraform entrypoint where the ParallelCluster module is configured.
### outputs.tf - Defines the cluster as a Terraform output.
### providers.tf - Configures the providers: ParallelCluster and AWS.
### terraform.tf - Import the ParallelCluster provider.
### terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
### variables.tf - Defines the variables, e.g. region, PCAPI stack name.

```


Distribuzione del cluster

Per distribuire il cluster, esegui i comandi Terraform standard nell'ordine.

Note

Questo esempio presuppone che tu abbia già distribuito l' ParallelCluster API nel tuo account.

1. Compila il progetto:

```
terraform init
```

2. Definisci il piano di implementazione:

```
terraform plan -out tfplan
```

3. Implementa il piano:

```
terraform apply tfplan
```

Implementa l' ParallelCluster API con i cluster

Se non hai distribuito l' ParallelCluster API e desideri distribuirla con i cluster, modifica i seguenti file:

- `main.tf`

```
module "pcluster" {  
  source = "aws-tf/aws/parallelcluster"  
  version = "1.0.0"  
  
  region          = var.region  
  api_stack_name  = var.api_stack_name  
  api_version     = var.api_version  
  deploy_pcluster_api = true  
  parameters = {  
    EnableIamAdminAccess = "true"  
  }  
  
  template_vars = local.config_vars
```

```
cluster_configs      = local.cluster_configs
config_path          = "config/clusters.yaml"
}
```

- providers.tf

```
provider "aws-parallelcluster" {
  region  = var.region
  profile = var.profile
  endpoint = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiInvokeUrl
  role_arn = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiUserRole
}
```

Autorizzazioni richieste

Sono necessarie le seguenti autorizzazioni per distribuire un cluster con Terraform:

- assume il ruolo dell' ParallelCluster API, che è responsabile dell'interazione con l'API ParallelCluster
- descrivi lo CloudFormation stack dell' ParallelCluster API per verificarne l'esistenza e recuperarne i parametri e gli output

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:sts::111122223333:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}
```

```
}  
]  
}
```

Creazione di un'AMI personalizzata con Terraform

Quando si utilizza AWS ParallelCluster, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [the section called “AWS servizi usati da AWS ParallelCluster”](#).

Prerequisiti

- Terraform v1.5.7+ è installato.
- [the section called “AWS ParallelCluster API”](#) v3.8.0+ è distribuito nel tuo account. Per informazioni, consulta [the section called “Creare un cluster con Terraform”](#).
- Ruolo IAM con le autorizzazioni per richiamare l'API. ParallelCluster Per informazioni, consulta [the section called “Autorizzazioni richieste”](#).

Definisci un progetto Terraform

In questo tutorial, definirai un semplice progetto Terraform per implementare un' ParallelCluster AMI personalizzata.

1. Crea una directory chiamata. `my-amis`

Tutti i file che creerai si troveranno all'interno di questa directory.

2. Crea il file `terraform.tf` per importare il ParallelCluster provider.

```
terraform {  
  required_version = ">= 1.5.7"  
  required_providers {  
    aws-parallelcluster = {  
      source = "aws-tf/aws-parallelcluster"  
      version = "~> 1.0"  
    }  
  }  
}
```

3. Crea il file `providers.tf` per configurare i AWS provider ParallelCluster e.

```

provider "aws" {
  region  = var.region
  profile = var.profile
}

provider "aws-parallelcluster" {
  region          = var.region
  profile         = var.profile
  api_stack_name = var.api_stack_name
  use_user_role   = true
}

```

4. Crea il file `main.tf` per definire le risorse utilizzando il ParallelCluster modulo.

Per esaminare le proprietà dell'immagine che è possibile impostare all'interno dell'`image_configuration` elemento, vedere [the section called “Crea file di configurazione delle immagini”](#).

Per esaminare le opzioni che è possibile impostare per la creazione di immagini, ad esempio `image_id` e `rollback_on_failure`, vedere [the section called “pcluster build-image”](#).

```

data "aws-parallelcluster_list_official_images" "parent_image" {
  region = var.region
  os      = var.os
  architecture = var.architecture
}

resource "aws-parallelcluster_image" "demo01" {
  image_id          = "demo01"
  image_configuration = yamlencode({
    "Build":{
      "InstanceType": "c5.2xlarge",
      "ParentImage": data.aws-
parallelcluster_list_official_images.parent_image.official_images[0].amiId,
      "UpdateOsPackages": {"Enabled": false}
    }
  })
  rollback_on_failure = false
}

```

5. Crea il file `variables.tf` per definire le variabili che possono essere inserite per questo progetto.

```
variable "region" {
  description = "The region the ParallelCluster API is deployed in."
  type        = string
  default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

variable "api_stack_name" {
  type        = string
  description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
  default     = "ParallelCluster"
}

variable "api_version" {
  type        = string
  description = "The version of the ParallelCluster API."
}

variable "os" {
  type        = string
  description = "The OS of the ParallelCluster image."
}

variable "architecture" {
  type        = string
  description = "The architecture of the ParallelCluster image."
}
```

6. Crea il file `terraform.tfvars` per impostare i tuoi valori arbitrari per le variabili.

Con il file seguente, distribuisce l'AMI personalizzata `us-east-1` basata sull'architettura Amazon Linux 2 per `x86_64`, utilizzando il `ParallelCluster API 3.11.1` esistente che è già distribuita con il nome dello stack `us-east-1 MyParallelClusterAPI-3111`

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-3111"
```

```
api_version = "3.11.1"

os = "alinux2"
architecture = "x86_64"
```

7. Crea il file per definire gli output restituiti `outputs.tf` da questo progetto.

```
output "parent_image" {
  value = data.aws-parallelcluster_list_official_images.parent_image.official_images[0]
}

output "custom_image" {
  value = aws-parallelcluster_image.demo01
}
```

La cartella del progetto è:

```
my-amis
### main.tf - Terraform entrypoint where the ParallelCluster module is configured.
### outputs.tf - Defines the cluster as a Terraform output.
### providers.tf - Configures the providers: ParallelCluster and AWS.
### terraform.tf - Import the ParallelCluster provider.
### terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
### variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

Implementa l'AMI

Per distribuire l'AMI, esegui i comandi Terraform standard nell'ordine.

1. Compila il progetto:

```
terraform init
```

2. Definisci il piano di implementazione:

```
terraform plan -out tfplan
```

3. Implementa il piano:

```
terraform apply tfplan
```

Autorizzazioni richieste

Sono necessarie le seguenti autorizzazioni per distribuire un'AMI personalizzata con Terraform:

- assume il ruolo dell' ParallelCluster API, che si occupa dell'interazione con l'API ParallelCluster
- descrivi lo CloudFormation stack dell' ParallelCluster API, per verificarne l'esistenza e recuperarne i parametri e gli output

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:sts::111122223333:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}
```

AWS ParallelCluster Integrazione dell'interfaccia utente con Identity Center

L'obiettivo di questo tutorial è dimostrare come integrare l' AWS ParallelCluster interfaccia utente con IAM Identity Center per una soluzione Single Sign-On che unifica gli utenti in Active Directory e può essere condivisa con i cluster. AWS ParallelCluster

Quando si utilizza AWS ParallelCluster, si pagano solo le AWS risorse create quando si creano o si AWS ParallelCluster aggiornano immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Prerequisiti:

- Un' AWS ParallelCluster interfaccia utente esistente che può essere installata seguendo le istruzioni riportate [qui](#).
- Un'Active Directory gestita esistente, preferibilmente una che utilizzerai anche per [l' AWS ParallelCluster integrazione](#).

Abilita IAM Identity Center

Se disponi già di un centro di identità collegato al tuo AWS Managed Microsoft AD (Active Directory), puoi utilizzarlo e passare alla sezione Aggiungere l'applicazione a IAM Identity Center.

Se non disponi già di un centro di identità collegato a un AWS Managed Microsoft AD, segui i passaggi seguenti per configurarlo.

Attivazione di Identity Center

1. Nella console, accedi a IAM Identity Center. (Assicurati di trovarti nella regione in cui hai il tuo AWS Managed Microsoft AD.)
2. Fai clic sul pulsante Abilita, potrebbe essere richiesto se desideri abilitare le organizzazioni. Questo è un requisito, quindi puoi scegliere di abilitarlo. Nota: in questo modo invierai all'amministratore del tuo account un'email di conferma che dovrai seguire il link per confermare.

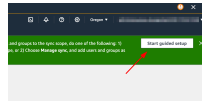
Connessione di Identity Center a Managed AD

1. Nella pagina successiva, dopo aver abilitato il centro di identità, dovresti vedere i passaggi di configurazione consigliati, nella fase 1, seleziona Scegli la tua origine di identità.

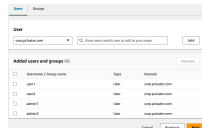
2. Nella sezione Origine dell'identità, fai clic sul menu a discesa Azioni (in alto a destra), quindi seleziona Cambia fonte di identità.
3. Seleziona Active Directory.
4. In Directory esistenti, scegli la tua directory.
5. Fare clic su Avanti.
6. Controlla le modifiche, scorri verso il basso, digita ACCETTA nella casella di testo per confermare, quindi fai clic su Cambia origine dell'identità.
7. Attendi il completamento delle modifiche, quindi dovresti vedere un banner verde in alto.

Sincronizzazione di utenti e gruppi con Identity Center

1. Nel banner verde, fai clic su Avvia configurazione guidata (pulsante in alto a destra)

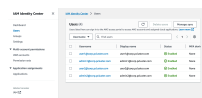


2. Nella sezione Configura le mappature degli attributi, fai clic su Avanti
3. Nella sezione Configura l'ambito di sincronizzazione, digita il nome degli utenti che desideri sincronizzare con Identity Center, quindi fai clic su Aggiungi
4. Una volta terminata l'aggiunta di utenti e gruppi, fai clic su Avanti



5. Controlla le modifiche, quindi fai clic su Salva configurazione
6. Se nella schermata successiva viene visualizzato un avviso relativo alla mancata sincronizzazione degli utenti, seleziona il pulsante Riprendi la sincronizzazione in alto a destra.
7. Successivamente, per abilitare gli utenti, nella scheda Utenti a sinistra, seleziona un utente, quindi fai clic su Abilita accesso utente > Abilita accesso utente

Nota: potrebbe essere necessario selezionare Riprendi sincronizzazione se hai un banner di avviso in alto e poi attendi che gli utenti si sincronizzino (prova il pulsante di aggiornamento per vedere se sono già sincronizzati).



Aggiungere l'applicazione a IAM Identity Center

Dopo aver sincronizzato gli utenti con IAM Identity Center, dovrai aggiungere una nuova applicazione. Questo configura quali applicazioni abilitate all'SSO saranno disponibili sul tuo portale IAM Identity Center. In questo caso, aggiungeremo l' AWS ParallelCluster interfaccia utente come applicazione mentre IAM Identity Center sarà il provider di identità.

Il passaggio successivo aggiungerà l' AWS ParallelCluster interfaccia utente come applicazione in IAM Identity Center. AWS ParallelCluster L'interfaccia utente è un portale web che aiuta l'utente a gestire i propri cluster. Per ulteriori informazioni, consulta [AWS ParallelCluster UI](#).

Configurazione dell'applicazione in Identity Center

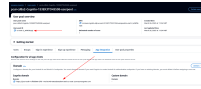
1. In IAM Identity Center > Applicazioni (nella barra del menu a sinistra, fai clic su Applicazioni)
2. Fai clic su Aggiungi applicazione
3. Seleziona Aggiungi un'applicazione SAML 2.0 personalizzata
4. Fai clic su Avanti
5. Seleziona il nome visualizzato e la descrizione che desideri utilizzare (ad esempio PCUI e AWS ParallelCluster UI)
6. Nella sezione Metadati IAM Identity Center, copia il link per il file di metadati IAM Identity Center SAML e salvalo per utilizzarlo in un secondo momento. Questo verrà utilizzato per configurare l'SSO sull'app web
7. In Proprietà dell'applicazione, nell'URL di avvio dell'applicazione, inserisci il tuo indirizzo PCUI. Puoi trovarlo accedendo alla CloudFormation console, selezionando lo stack che corrisponde a PCUI (ad esempio parallelcluster-ui) e andando alla scheda Output per trovare ParallelCluster UIUrl

ad esempio `https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com`

8. In Metadati dell'applicazione, scegli Digita manualmente i valori dei metadati. Fornisci quindi i seguenti valori.
 - a. Importante: assicurati di sostituire i valori domain-prefix, region e userpool-id con informazioni specifiche del tuo ambiente.
 - b. Il prefisso del dominio, la regione e l'userpool-id possono essere ottenuti aprendo la console Amazon Cognito > User pools



- c. Seleziona il pool di utenti che corrisponde a PCUI (che avrà un nome di pool di utenti come PCUI-CD8A2-Cognito-153 EK3 TO45 S98-UserPool)
- d. Vai a Integrazione delle app



- 9. <domain-prefix>URL dell'Application Assertion Consumer Service (ACS): https://.auth.<region>.amazoncognito.com/saml2/idpresponse

Destinatari SAML dell'applicazione: urn:amazon:cognito:sp: <userpool-id>

- 10. Seleziona Invia. Quindi, vai alla pagina dei dettagli dell'applicazione che hai aggiunto.
- 11. Seleziona l'elenco a discesa Azioni e scegli Modifica mappature degli attributi. Quindi, fornisci i seguenti attributi.
 - a. Attributo utente nell'applicazione: soggetto (Nota: l'oggetto è precompilato). → Esegue il mapping a questo valore di stringa o attributo utente in IAM Identity Center: \$ {user:email}, Format: emailAddress
 - b. Attributo utente nell'applicazione: email → Esegue il mapping a questo valore di stringa o attributo utente in IAM Identity Center: \$ {user:email}, Formato: unspecified



- 12. Salvare le modifiche.
- 13. Scegli il pulsante Assegna utenti, quindi assegna l'utente all'applicazione. Questi sono gli utenti di Active Directory che avranno accesso all'interfaccia PCUI.



Configura IAM Identity Center come IdP SAML nel tuo pool di utenti

- 1. Nelle impostazioni del pool di utenti, seleziona Esperienza di accesso > Aggiungi provider di identità



- 2. Scegli un IdP SAML
- 3. Per il nome del provider, fornisci IdentityCenter

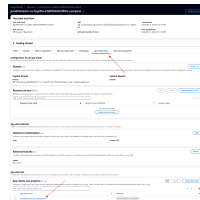
4. In Origine del documento di metadati scegli Inserisci l'URL dell'endpoint del documento di metadati e fornisci l'URL copiato durante la configurazione dell'applicazione di Identity Center
5. In Attributi, per e-mail scegli email



6. Seleziona Aggiungi provider di identità.

Integra l'IdP con il client dell'app del pool di utenti

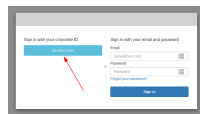
1. Successivamente, nella sezione Integrazione delle app del tuo pool di utenti, scegli il client elencato nell'elenco dei client dell'app



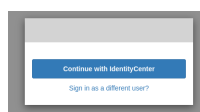
2. In Hosted UI scegli Modifica
3. Nella sezione Provider di identità scegli IdentityCenteranche.
4. Scegliere Salva modifiche.

Convalida la configurazione

1. Successivamente convalideremo la configurazione che abbiamo appena creato accedendo a PCUI. Accedi al tuo portale PCUI e ora dovresti vedere un'opzione per accedere con il tuo ID aziendale:



2. Facendo clic sul IdentityCenterpulsante dovresti accedere al login IAM Identity Center IdP seguito da una pagina con le tue applicazioni che include PCUI, apri quell'applicazione.
3. Quando arrivi alla schermata seguente, l'utente sarà stato aggiunto al pool di utenti di Cognito.



Rendi il tuo utente un amministratore

1. Ora vai alla console Amazon Cognito > Pool di utenti e seleziona l'utente appena creato, che dovrebbe avere il prefisso identitycenter.



2. In Appartenenze al gruppo seleziona Aggiungi utente al gruppo, scegli amministratore e fai clic su Aggiungi.
3. Ora, quando fai clic su Continua con, IdentityCenter verrai indirizzato alla pagina dell' AWS ParallelCluster interfaccia utente.

Esecuzione di lavori containerizzati con Pyxis

Scopri come creare un cluster in grado di eseguire lavori containerizzati utilizzando Pyxis, un plug-in SPANK per gestire i lavori containerizzati in SLURM. I contenitori in Pyxis sono gestiti da Enroot, uno strumento per trasformare le immagini tradizionali in sandbox non privilegiate. container/OS [Per ulteriori informazioni, consulta NVIDIA Pyxis e NVIDIA Enroot.](#)

Note

- Questa funzionalità è disponibile con la versione 3.11.1 AWS ParallelCluster
- Gli script di questo tutorial spostano (mv) alcuni file, eliminandoli dalle loro posizioni originali. Se volete conservare le copie di questi file nelle loro posizioni originali, modificate gli script per usare invece il comando copy (cp).

Quando si utilizza AWS ParallelCluster, si pagano solo le AWS risorse create quando si creano o si aggiornano AWS ParallelCluster immagini e cluster. Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Prerequisiti:

- AWS CLI È [installato e configurato](#).
- Una [coppia di chiavi Amazon EC2](#).
- Un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [CLI di pcluster](#).

Creazione del cluster

A partire dalla AWS ParallelCluster 3.11.1, tutte le versioni ufficiali AMIs vengono fornite con Pyxis ed Enroot preinstallati. In particolare, SLURM viene ricompilato con il supporto Pyxis ed Enroot viene installato come binario nel sistema. Tuttavia, è necessario configurarli in base alle proprie esigenze specifiche. Le cartelle utilizzate da Enroot e Pyxis avranno un impatto critico sulle prestazioni del cluster. [Per ulteriori informazioni, consulta la documentazione di Pyxis e la documentazione di Enroot.](#)

Per comodità, puoi trovare configurazioni di esempio per Pyxis, Enroot e SPANK all'interno. `/opt/parallelcluster/examples/`

Per distribuire un cluster utilizzando le configurazioni di esempio che abbiamo fornito, completa il seguente tutorial.

Per creare il cluster con una configurazione di esempio

Pyxis ed Enroot devono essere configurati sul nodo principale creando prima le directory persistenti e volatili per Enroot, quindi creando la directory di runtime per Pyxis e infine abilitando Pyxis come plugin SPANK nell'intero cluster.

1. Esegui lo script seguente come azione [OnNodeConfigured](#) personalizzata nel nodo principale per configurare Pyxis ed Enroot sul nodo principale.

```
#!/bin/bash
set -e

echo "Executing $0"

# Configure Enroot
ENROOT_PERSISTENT_DIR="/var/enroot"
ENROOT_VOLATILE_DIR="/run/enroot"

sudo mkdir -p $ENROOT_PERSISTENT_DIR
sudo chmod 1777 $ENROOT_PERSISTENT_DIR
sudo mkdir -p $ENROOT_VOLATILE_DIR
sudo chmod 1777 $ENROOT_VOLATILE_DIR
sudo mv /opt/parallelcluster/examples/enroot/enroot.conf /etc/enroot/enroot.conf
sudo chmod 0644 /etc/enroot/enroot.conf

# Configure Pyxis
PYXIS_RUNTIME_DIR="/run/pyxis"
```

```
sudo mkdir -p $PYXIS_RUNTIME_DIR
sudo chmod 1777 $PYXIS_RUNTIME_DIR

sudo mkdir -p /opt/slurm/etc/plugstack.conf.d/
sudo mv /opt/parallelcluster/examples/spank/plugstack.conf /opt/slurm/etc/
sudo mv /opt/parallelcluster/examples/pyxis/pyxis.conf /opt/slurm/etc/
plugstack.conf.d/
sudo -i scontrol reconfigure
```

2. Pyxis ed Enroot devono essere configurati sulla flotta di calcolo creando le directory persistenti e volatili per Enroot e la directory di runtime per Pyxis. Esegui lo script seguente come azione [OnNodeStart](#) personalizzata nei nodi di calcolo per configurare Pyxis ed Enroot sulla flotta di elaborazione.

```
#!/bin/bash
set -e

echo "Executing $0"

# Configure Enroot
ENROOT_PERSISTENT_DIR="/var/enroot"
ENROOT_VOLATILE_DIR="/run/enroot"
ENROOT_CONF_DIR="/etc/enroot"

sudo mkdir -p $ENROOT_PERSISTENT_DIR
sudo chmod 1777 $ENROOT_PERSISTENT_DIR
sudo mkdir -p $ENROOT_VOLATILE_DIR
sudo chmod 1777 $ENROOT_VOLATILE_DIR
sudo mkdir -p $ENROOT_CONF_DIR
sudo chmod 1777 $ENROOT_CONF_DIR
sudo mv /opt/parallelcluster/examples/enroot/enroot.conf /etc/enroot/enroot.conf
sudo chmod 0644 /etc/enroot/enroot.conf

# Configure Pyxis
PYXIS_RUNTIME_DIR="/run/pyxis"

sudo mkdir -p $PYXIS_RUNTIME_DIR
sudo chmod 1777 $PYXIS_RUNTIME_DIR

# In Ubuntu24.04 Apparmor blocks the creation of unprivileged user namespaces,
```

```
# which is required by Enroot. So to run Enroot, it is required to disable this
restriction.
# See https://ubuntu.com/blog/ubuntu-23-10-restricted-unprivileged-user-namespaces
source /etc/os-release
if [ "${ID}${VERSION_ID}" == "ubuntu24.04" ]; then
    echo "kernel.apparmor_restrict_unprivileged_userns = 0" | sudo tee /etc/
sysctl.d/99-pcluster-disable-apparmor-restrict-unprivileged-userns.conf
    sudo sysctl --system
fi
```

Invia offerte di lavoro

Ora che Pyxis è configurato nel cluster, è possibile inviare lavori containerizzati utilizzando i comandi sbatch e srun, che ora sono arricchiti con opzioni specifiche del contenitore.

```
# Submitting an interactive job
srun -N 2 --container-image docker://ubuntu:22.04 hostname

# Submitting a batch job
sbatch -N 2 --wrap='srun --container-image docker://ubuntu:22.04 hostname'
```

Creazione di un cluster con un Lustre abilitato per EFA FSx

In questo tutorial, creerai un cluster che utilizza un file system FSx Lustre abilitato all'EFA come storage condiviso. L'utilizzo di un file system FSx Lustre con EFA abilitato può fornire un incremento delle prestazioni fino a 8 volte. Per verificare se un file system compatibile [con EFA è ciò di cui hai bisogno, consulta la sezione Utilizzo dei file system abilitati per EFA nella Guida per l'utente di for Lustre](#). FSx

Quando si utilizza AWS ParallelCluster, si pagano solo AWS le risorse create durante la creazione o l'aggiornamento di immagini e cluster. AWS ParallelCluster Per ulteriori informazioni, consulta [AWS servizi usati da AWS ParallelCluster](#).

Requisiti

- La AWS CLI è [installata e](#) configurata.
- La ParallelCluster CLI è [installata e](#) configurata.

- Una [coppia di chiavi Amazon EC2](#) per accedere al cluster.
- Un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la ParallelCluster CLI.

Creazione di gruppi di sicurezza

Crea due gruppi di sicurezza nello stesso VPC in cui verranno distribuiti il cluster e il file system: uno per il client in esecuzione sui nodi del cluster e uno per il file system.

```
# Create security group for the FSx client
aws ec2 create-security-group \
  --group-name Fsx-Client-SecurityGroup \
  --description "Allow traffic for the FSx Lustre client" \
  --vpc-id vpc-cluster \
  --region region

# Create security group for the FSx file system
aws ec2 create-security-group \
  --group-name Fsx-Filesystem-SecurityGroup \
  --description "Allow traffic for the FSx Lustre File System" \
  --vpc-id vpc-cluster \
  --region region
```

Nel resto del tutorial, assumeremo `sg-client` e `sg-file-system` siano gli ID dei gruppi di sicurezza rispettivamente del client e del file system.

Configura il gruppo di sicurezza per il client per consentire tutto il traffico in uscita verso il file system, come [richiesto da](#) EFA.

```
# Allow all outbound traffic from the client to the file system
aws ec2 authorize-security-group-egress \
  --group-id sg-client \
  --protocol -1 \
  --port -1 \
  --source-group sg-file-system \
  --region region
```

Configura il gruppo di sicurezza per il file system per consentire tutto il inbound/outbound traffico interno e tutto il traffico in entrata dal client, come [richiesto](#) dall'EFA.

```
# Allow all inbound traffic within this security group
aws ec2 authorize-security-group-ingress \
```

```
--group-id sg-file-system \  
--protocol -1 \  
--port -1 \  
--source-group sg-file-system \  
--region region  
  
# Allow all outbound traffic within this security group  
aws ec2 authorize-security-group-egress \  
  --group-id sg-file-system \  
  --protocol -1 \  
  --port -1 \  
  --source-group sg-file-system \  
  --region region  
  
# Allow all inbound traffic from the client  
aws ec2 authorize-security-group-ingress \  
  --group-id sg-file-system \  
  --protocol -1 \  
  --port -1 \  
  --source-group sg-client \  
  --region region  
  
# Allow all outbound traffic to the client  
aws ec2 authorize-security-group-egress \  
  --group-id sg-file-system \  
  --protocol -1 \  
  --port -1 \  
  --source-group sg-client \  
  --region region
```

Crea il file system

Crea il file system all'interno della stessa zona di disponibilità (AZ) in cui si troveranno i nodi di elaborazione e *subnet-compute-nodes* sostituiscilo con il relativo ID nel codice seguente. Ciò è necessario per consentire a EFA di funzionare con il file system. Nota che, come parte della creazione del file system, abilitiamo EFA utilizzando la `EfaEnable` proprietà.

```
aws fsx create-file-system \  
  --file-system-type LUSTRE \  
  --storage-capacity 38400 \  
  --storage-type SSD \  
  --subnet-ids subnet-compute-nodes \  
  --security-group-ids sg-file-system \  
  --tags Tag
```

```
--lustre-configuration
DeploymentType=PERSISTENT_2,PerUnitStorageThroughput=125,EfaEnabled=true,MetadataConfiguration
\
--region region
```

Prendete nota dell'id del file system restituito dal comando precedente. Nel resto del tutorial, sostituisco *fs-id* con questo id del file system.

Creazione del cluster

1. Crea il cluster con le seguenti configurazioni impostate nel file di configurazione AWS ParallelCluster YAML:
 - a. AMI basata su un sistema operativo supportato, come Ubuntu 22.04.
 - b. I nodi di calcolo devono utilizzare un [tipo di istanza supportato da EFA con Nitro v4+, ad esempio](#) g6.16xlarge.
 - I nodi di calcolo devono trovarsi nella stessa AZ in cui si trova il file system.
 - I nodi di calcolo devono avere [Efa/Enabled](#) impostato su true.
 - I nodi di calcolo devono eseguire lo script di configurazione `configure-efa-fsx-lustre-client.sh` come azione personalizzata. [OnNodeStart](#) Lo script, fornito nella [documentazione FSx ufficiale](#) e offerto nel nostro bucket pubblico per comodità, ha lo scopo di configurare il client FSx Lustre sui nodi di calcolo per consentire loro di utilizzare EFA.
2. Crea un file di configurazione del cluster: `config.yaml`

```
Region: region
Image:
  Os: ubuntu2204
HeadNode:
  InstanceType: c5.xlarge
  Networking:
    SubnetId: subnet-xxxxxxxxxx
    AdditionalSecurityGroups:
      - sg-client
  Ssh:
    KeyName: my-ssh-key
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: q1
    ComputeResources:
```

```

- Name: cr1
  Instances:
    - InstanceType: g6.16xlarge
  MinCount: 1
  MaxCount: 3
  Efa:
    Enabled: true
  Networking:
    SubnetIds:
      - subnet-xxxxxxxxxx # Subnet in the same AZ where the file system is
  AdditionalSecurityGroups:
    - sg-client
  PlacementGroup:
    Enabled: false
  CustomActions:
    OnNodeStart:
      Script: https://us-east-1-aws-parallelcluster.s3.us-east-1.amazonaws.com/
scripts/fsx-lustre-efa/configure-efa-fsx-lustre-client.sh
  SharedStorage:
    - MountDir: /fsx
      Name: my-fsxlustre-efa-external
      StorageType: FsxLustre
      FsxLustreSettings:
        FileSystemId: fs-id

```

Quindi crea un cluster utilizzando quella configurazione:

```

pcluster create-cluster \
  --cluster-name fsx-efa-tutorial \
  --cluster-configuration config.yaml \
  --region region

```

La convalida FSx con EFA funziona

Per verificare che il traffico di rete Lustre utilizzi EFA, utilizzate `lnetctl` lo strumento Lustre in grado di mostrare il traffico di rete per una determinata interfaccia di rete. A tal fine, esegui i seguenti comandi in un nodo di calcolo:

```

# Take note of the number of packets flowing through the interface,
# which are specified in statistics:send_count and statistics:recv_count
sudo lnetctl net show --net efa -v

```

```
# Generate traffic to the file system
echo 'Hello World' > /fsx/hello-world.txt

# Take note of the number of packets flowing through the interface,
# which are specified in statistics:send_count and statistics:recv_count
sudo lnctl net show --net efa -v
```

Se la funzionalità funziona, si prevede che il numero di pacchetti che fluiscono attraverso l'interfaccia aumenti.

Supporta Nvidia-IMEX con istanza p6e-gb200

Questo tutorial mostra come iniziare a usare P6e-GB200, per sfruttare le AWS ParallelCluster massime prestazioni della GPU per l'addestramento e l'inferenza dell'IA. Le [istanze p6e-gb200.36xlarge sono disponibili solo tramite P6e](#), dove è Ultraserver Size ed è la dimensione che forma l'ultraserver. GB200 UltraServers u-p6e-gb200x72 p6e-gb200.36xlarge [InstanceType Al momento dell'acquisto, un Ultraserver sarà disponibile tramite un EC2 Capacity u-p6e-gb200x72 Blocks for ML che avrà 18 istanze.](#) p6e-gb200.36xlarge [Per ulteriori informazioni, consulta P6e-GB200](#)

AWS ParallelCluster versione 3.14.0:

- fornisce lo stack software NVIDIA completo (driver, CUDA, EFA, NVIDIA-IMEX) richiesto da questo tipo di istanza
- crea configurazioni GB200 nvidia-imex per ultraserver P6e-
- abilita e avvia il servizio per nvidia-imex P6e- ultraserver GB200
- configura il plug-in di topologia Slurm Block in modo che ogni Ultraserver P6e-GB200 (un blocco di capacità EC2) sia uno Slurm Block della giusta dimensione (vedi la voce relativa alla versione 3.14.0). [Note di rilascio e cronologia dei documenti](#)

Tuttavia, GPU-to-GPU la comunicazione via cavo NVLink richiede configurazioni aggiuntive, in particolare un [nodes_config.cfg](#) file che contiene gli indirizzi IP dei nodi di calcolo in un dominio IMEX che non viene generato automaticamente. ParallelCluster Per facilitare la generazione di questo file, forniamo uno script prolog che rileva automaticamente il nodo di calcolo IPs e configura i [nodes_config.cfg](#) seguenti consigli di [NVIDIA IMEX Slurm](#) Job Scheduler Integration. Questo tutorial illustra come creare lo script prolog, distribuirlo tramite un'azione personalizzata e convalidare la configurazione IMEX. HeadNode

Note

P6e- GB200 è supportato a partire dalla versione AWS ParallelCluster 3.14.0 su Amazon Linux 2023, Ubuntu 22.04 e Ubuntu 24.04. [Per le versioni software dettagliate e un elenco aggiornato delle distribuzioni supportate, consulta il changelog.AWS ParallelCluster](#)

Crea uno script Prolog per gestire Nvidia-IMEX

Limitazione:

- Questo script di prolog verrà eseguito al momento dell'invio di un lavoro esclusivo. Questo serve a garantire che un riavvio IMEX non interrompa i processi in esecuzione sui nodi P6e-GB200 che appartengono a un dominio IMEX.

Di seguito è riportato `91_nvidia_imex_prolog.sh` lo script da configurare come prolog in Slurm. Viene utilizzato per aggiornare automaticamente la configurazione `nvidia-imex` sui nodi di calcolo.

[Il nome dello script ha il prefisso di 91 aderire alla convenzione di denominazione di SchedMD.](#)

Ciò garantisce che venga eseguito prima di qualsiasi altro script di prolog nella sequenza. Lo script riconfigura la configurazione del nodo NVIDIA Imex all'avvio di un processo e ricarica i daemon NVIDIA necessari.

Note

Questo script non verrà eseguito nel caso in cui più lavori vengano avviati contemporaneamente sugli stessi nodi, pertanto suggeriamo di utilizzare il flag all'invio. `--exclusive`

```
#!/usr/bin/env bash

# This prolog script configures the NVIDIA IMEX on compute nodes involved in the job
# execution.
#
# In particular:
# - Checks whether the job is executed exclusively.
#   If not, it exits immediately because it requires jobs to be executed exclusively.
# - Checks if it is running on a p6e-gb200 instance type.
```

```

# If not, it exits immediately because IMEX must be configured only on that instance
type.
# - Checks if the IMEX service is enabled.
# If not, it exits immediately because IMEX must be enabled to get configured.
# - Creates the IMEX default channel.
# For more information about IMEX channels, see https://docs.nvidia.com/multi-node-nvlink-systems/imex-guide/imexchannels.html
# - Writes the private IP addresses of compute nodes into /etc/nvidia-imex/
nodes_config.cfg.
# - Restarts the IMEX system service.
#
# REQUIREMENTS:
# - This prolog assumes to be run only with exclusive jobs.

LOG_FILE_PATH="/var/log/parallelcluster/nvidia-imex-prolog.log"
SCONTROL_CMD="/opt/slurm/bin/scontrol"
IMEX_START_TIMEOUT=60
IMEX_STOP_TIMEOUT=15
ALLOWED_INSTANCE_TYPES="^(p6e-gb200)"
IMEX_SERVICE="nvidia-imex"
IMEX_NODES_CONFIG="/etc/nvidia-imex/nodes_config.cfg"

function info() {
    echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [INFO] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function warn() {
    echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [WARN] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function error() {
    echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [ERROR] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function error_exit() {
    error "$1" && exit 1
}

function prolog_end() {
    info "PROLOG End JobId=${SLURM_JOB_ID}: $0"
    info "-----"
    exit 0
}

```

```

function get_instance_type() {
    local token=$(curl -X PUT -s "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600")
    curl -s -H "X-aws-ec2-metadata-token: ${token}" http://169.254.169.254/latest/meta-
data/instance-type
}

function return_if_unsupported_instance_type() {
    local instance_type=$(get_instance_type)

    if [[ ! ${instance_type} =~ ${ALLOWED_INSTANCE_TYPES} ]]; then
        info "Skipping IMEX configuration because instance type ${instance_type} does not
support it"
        prolog_end
    fi
}

function return_if_imex_disabled() {
    if ! systemctl is-enabled "${IMEX_SERVICE}" &>/dev/null; then
        warn "Skipping IMEX configuration because system service ${IMEX_SERVICE} is not
enabled"
        prolog_end
    fi
}

function return_if_job_is_not_exclusive() {
    if [[ "${SLURM_JOB_OVERSUBSCRIBE}" =~ ^(NO|TOP0)$ ]]; then
        info "Job is exclusive, proceeding with IMEX configuration"
    else
        info "Skipping IMEX configuration because the job is not exclusive"
        prolog_end
    fi
}

function get_ips_from_node_names() {
    local _nodes=$1
    ${SCONTROL_CMD} -ao show node "${_nodes}" | sed 's/^.* NodeAddr=\([^ ]*\)\.*\1/'
}

function get_compute_resource_name() {
    local _queue_name_prefix=$1
    local _slurmd_node_name=$2
    echo "${_slurmd_node_name}" | sed -E "s/${_queue_name_prefix}(.)-[0-9]+$/\1/"
}

```



```

function reload_imex() {
    info "Stopping IMEX"
    timeout ${IMEX_STOP_TIMEOUT} systemctl stop ${IMEX_SERVICE}
    pkill -9 ${IMEX_SERVICE}

    info "Restarting IMEX"
    timeout ${IMEX_START_TIMEOUT} systemctl start ${IMEX_SERVICE}
}

function create_default_imex_channel() {
    info "Creating IMEX default channel"
    MAJOR_NUMBER=$(cat /proc/devices | grep nvidia-caps-imex-channels | cut -d' ' -f1)
    if [ ! -d "/dev/nvidia-caps-imex-channels" ]; then
        sudo mkdir /dev/nvidia-caps-imex-channels
    fi

    # Then check and create device node
    if [ ! -e "/dev/nvidia-caps-imex-channels/channel0" ]; then
        sudo mknod /dev/nvidia-caps-imex-channels/channel0 c $MAJOR_NUMBER 0
        info "IMEX default channel created"
    else
        info "IMEX default channel already exists"
    fi
}

{
    info "PROLOG Start JobId=${SLURM_JOB_ID}: $0"

    return_if_job_is_not_exclusive
    return_if_unsupported_instance_type
    return_if_imex_disabled

    create_default_imex_channel

    IPS_FROM_CR=$(get_ips_from_node_names "${SLURM_NODELIST}")

    info "Node Names: ${SLURM_NODELIST}"
    info "Node IPs: ${IPS_FROM_CR}"
    info "IMEX Nodes Config: ${IMEX_NODES_CONFIG}"

    info "Updating IMEX nodes config ${IMEX_NODES_CONFIG}"
    echo "${IPS_FROM_CR}" > "${IMEX_NODES_CONFIG}"
    reload_imex
}

```

```

prolog_end

} 2>&1 | tee -a "${LOG_FILE_PATH}" | logger -t "91_nvidia_imex_prolog"

```

Crea lo script di HeadNode OnNodeStart azione personalizzato

Crea un'azione `install_custom_action.sh` personalizzata che scaricherà il suddetto script `prolog` in una directory condivisa a `/opt/slurm/etc/scripts/prolog.d/` cui accede Compute Nodes e imposterà le autorizzazioni appropriate per l'esecuzione.

```

#!/bin/bash
set -e

echo "Executing $0"

PROLOG_NVIDIA_IMEX=/opt/slurm/etc/scripts/prolog.d/91_nvidia_imex_prolog.sh
aws s3 cp "s3://<Bucket>/91_nvidia_imex_prolog.sh" "${PROLOG_NVIDIA_IMEX}"
chmod 0755 "${PROLOG_NVIDIA_IMEX}"

```

Creazione del cluster

Crea un cluster che includa istanze P6e-. GB200 Di seguito è riportato un esempio di configurazione contenente SlurmQueues il tipo Ultraserver. `u-p6e-gb200x72`

P6e- GB200 è attualmente disponibile solo nelle Local Zones. Alcune [Local Zones non supportano un gateway NAT](#), quindi segui le [opzioni di connettività per Local Zones](#) se ParallelCluster necessario per connetterti [Configurazione di gruppi di sicurezza per ambienti con restrizioni](#) ai AWS Servizi. Segui [Avvia istanze con Capacity Blocks \(CB\)](#) (AWS ParallelClusterLaunch) poiché gli Ultraserver sono disponibili solo come blocchi di capacità.

```

HeadNode:
  CustomActions:
    OnNodeStart:
      Script: s3://<s3-bucket-name>/install_custom_action.sh
  S3Access:
    - BucketName: <s3-bucket-name>
  InstanceType: <HeadNode-instance-type>
  Networking:
    SubnetId: <subnet-abcd78901234567890>

```

```

Ssh:
  KeyName: <Key-name>
Image:
  Os: ubuntu2404
Scheduling:
  Scheduler: slurm
  SlurmSettings:
    CustomSlurmSettings:
      - PrologFlags: "Alloc,NoHold"
      - MessageTimeout: 240
  SlurmQueues:
    - CapacityReservationTarget:
        CapacityReservationId: <cr-123456789012345678>
        CapacityType: CAPACITY_BLOCK
        ComputeResources: ### u-p6e-gb200x72
        - DisableSimultaneousMultithreading: true
        Efa:
          Enabled: true
          InstanceType: p6e-gb200.36xlarge
          MaxCount: 18
          MinCount: 18
          Name: cr1
        Name: q1
      Networking:
        SubnetIds:
          - <subnet-1234567890123456>

```

Convalida la configurazione IMEX

Il `91_nvidia_imex_prolog.sh` prolog verrà eseguito quando invii un lavoro Slurm. Di seguito è riportato un esempio di lavoro per verificare lo stato del dominio Nvidia-IMEX.

```

#!/bin/bash
#SBATCH --job-name=nvidia-imex-status-job
#SBATCH --ntasks-per-node=1
#SBATCH --output=slurm-%j.out
#SBATCH --error=slurm-%j.err

QUEUE_NAME="q1"
COMPUTE_RES_NAME="cr1"
IMEX_CONFIG_FILE="/opt/parallelcluster/shared/nvidia-imex/config_${QUEUE_NAME}_
${COMPUTE_RES_NAME}.cfg"

```

```
srun bash -c "/usr/bin/nvidia-imex-ctl -N -c ${IMEX_CONFIG_FILE} > result_\n${SLURM_JOB_ID}_\$(hostname).out 2> result_\${SLURM_JOB_ID}_\$(hostname).err"
```

Controlla l'output del Job:

Connectivity Table Legend:
I - Invalid - Node wasn't reachable, no connection status available
N - Never Connected
R - Recovering - Connection was lost, but clean up has not yet been triggered.
D - Disconnected - Connection was lost, and clean up has been triggreed.
A - Authenticating - If GSSAPI enabled, client has initiated mutual authentication.
!V! - Version mismatch, communication disabled.
!M! - Node map mismatch, communication disabled.
C - Connected - Ready for operation

5/12/2025 06:08:10.580

Nodes:

Node #0	-	172.31.48.81	-	READY	-	Version: 570.172
Node #1	-	172.31.48.98	-	READY	-	Version: 570.172
Node #2	-	172.31.48.221	-	READY	-	Version: 570.172
Node #3	-	172.31.49.228	-	READY	-	Version: 570.172
Node #4	-	172.31.50.39	-	READY	-	Version: 570.172
Node #5	-	172.31.50.44	-	READY	-	Version: 570.172
Node #6	-	172.31.51.66	-	READY	-	Version: 570.172
Node #7	-	172.31.51.157	-	READY	-	Version: 570.172
Node #8	-	172.31.52.239	-	READY	-	Version: 570.172
Node #9	-	172.31.53.80	-	READY	-	Version: 570.172
Node #10	-	172.31.54.95	-	READY	-	Version: 570.172
Node #11	-	172.31.54.183	-	READY	-	Version: 570.172
Node #12	-	172.31.54.203	-	READY	-	Version: 570.172
Node #13	-	172.31.54.241	-	READY	-	Version: 570.172
Node #14	-	172.31.55.59	-	READY	-	Version: 570.172
Node #15	-	172.31.55.187	-	READY	-	Version: 570.172
Node #16	-	172.31.55.197	-	READY	-	Version: 570.172
Node #17	-	172.31.56.47	-	READY	-	Version: 570.172

Nodes	From\To	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	0	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
	1	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
	2	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
	3	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
	4	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
	5	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

6	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
7	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
8	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
9	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
10	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
11	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
12	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
13	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
14	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
15	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
16	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
17	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

Domain State: UP

Personalizza le interfacce di rete dei nodi di calcolo con sostituzioni dei modelli di avvio

A partire dalla AWS ParallelCluster versione 3.15.0, il `LaunchTemplateOverrides` parametro consente di personalizzare le interfacce di rete dei nodi di calcolo sovrascrivendo la configurazione dell'interfaccia di rete predefinita con la configurazione in un modello di avvio di riferimento. L'intera sezione dell'interfaccia di rete dei nodi di calcolo viene sovrascritta dalla sezione dell'interfaccia di rete del modello di avvio utilizzato per l'override.

Questo tutorial illustra un esempio di sovrascrittura della configurazione di rete predefinita dei nodi di calcolo. `p6-b300.48xlarge` Questa personalizzazione è utile quando è necessaria una configurazione specifica dell'interfaccia di rete diversa da quella AWS ParallelCluster predefinita.

[In questo esempio, configuriamo lo use case 2 per le istanze P6-B300 come indicato nella documentazione dei tipi di istanze supportati da Amazon EC2 EFA.](#)

Note

Si consiglia di utilizzare per creare il modello di avvio AWS CLI anziché la console per la massima flessibilità.

 Note

Il modello di avvio deve contenere solo le sostituzioni delle interfacce di rete. AWS ParallelCluster ha una convalida che impedisce la sovrascrittura di altri parametri.

 Warning

Se si utilizza l'override per configurare le interfacce di rete in un modo non supportato dal tipo di istanza utilizzato, le istanze non verranno avviate.

Prerequisiti

- AWS ParallelCluster [è installata la versione 3.15.0 o successiva](#).
- AWS CLI [è installato e configurato](#).
- Hai un ruolo IAM con le [autorizzazioni](#) necessarie per eseguire la [pcluster](#) CLI.

Fase 1: Creare gruppi di sicurezza

Quando si crea il modello di avvio da utilizzare nell'override, è necessario fare riferimento a un gruppo di sicurezza. Il gruppo AWS ParallelCluster di sicurezza predefinito per la risorsa di calcolo non esiste fino alla creazione del cluster, quindi è necessario creare un gruppo di sicurezza personalizzato. Questo gruppo di sicurezza deve quindi essere referenziato dal gruppo di sicurezza del nodo principale per consentire il traffico tra il nodo principale e i nodi di elaborazione.

Se stai aggiornando un cluster esistente per personalizzare una nuova capacità, puoi utilizzare il gruppo di sicurezza del nodo di AWS ParallelCluster calcolo predefinito nel modello di avvio anziché crearne uno personalizzato.

Crea i due gruppi di sicurezza seguenti:

- Gruppo di sicurezza aggiuntivo del nodo principale (sg-1234abcd):
 - Ingresso: tutto il traffico proveniente dal gruppo di sicurezza di elaborazione
- Gruppo di sicurezza di calcolo (): sg-abcd1234
 - Ingresso: tutto il traffico proveniente dal gruppo di sicurezza del nodo principale
 - Ingresso: tutto il traffico proveniente da self () compute-to-compute

- Uscita: impostazione predefinita allow-all

Fase 2: Creare il modello di lancio

Crea un modello di avvio che definisca la configurazione dell'interfaccia di rete per i nodi di p6-b300.48xlarge calcolo. Per l'interfaccia di rete principale (indice della scheda di rete 0, indice del dispositivo 0), utilizzate un'interfaccia di rete ENA (predefinita). Per le altre schede di rete, create un'interfaccia solo EFA (indici delle schede di rete 1-16, indice del dispositivo 0) e un'interfaccia ENA (predefinita) (indici delle schede di rete 1-16, indice del dispositivo 1).

Eseguite il seguente AWS CLI comando per creare il modello di avvio (): lt-123456789

```
aws ec2 create-launch-template \
  --region us-east-1 \
  --launch-template-name override-lt \
  --launch-template-data '{
    "NetworkInterfaces": [
      {"NetworkCardIndex":0, "DeviceIndex":0, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":1, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":1, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":2, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":2, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":3, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":3, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":4, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":4, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":5, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":5, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
      {"NetworkCardIndex":6, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
```

```

    {"NetworkCardIndex":6, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":7, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":7, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":8, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":8, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":9, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":9, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":10, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":10, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":11, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":11, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":12, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":12, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":13, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":13, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":14, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":14, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":15, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":15, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":16, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":16, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"}
]

```



```
}'
```

Passaggio 3: Creare il cluster con le sostituzioni del modello di avvio

Crea una configurazione del cluster che utilizzi il `LaunchTemplateOverrides` parametro per fare riferimento al modello di lancio che hai creato.

```
Region: us-east-1
HeadNode:
  InstanceType: c5.xlarge
  Networking:
    SubnetId: subnet-abcdefghi
    AdditionalSecurityGroups:
      # Add the head node SG that allows traffic from the compute node SG
      - sg-1234abcd
  ...

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
    Networking:
      SubnetIds:
        - subnet-123456789
    ComputeResources:
      - Name: compute-resource1
        InstanceType: p6-b300.48xlarge
        Efa:
          Enabled: false # The override replaces all network interface configuration,
so this setting is ignored
        LaunchTemplateOverrides:
          LaunchTemplateId: lt-123456789
          Version: 1 # If the launch template is updated, then the new version should
be specified here.
```

AWS ParallelCluster risoluzione dei problemi

Le seguenti sezioni forniscono suggerimenti per la risoluzione dei problemi che potrebbero verificarsi durante l'utilizzo AWS ParallelCluster. La AWS ParallelCluster community mantiene una pagina Wiki che fornisce molti suggerimenti per la risoluzione dei problemi sul [AWS ParallelCluster GitHub Wiki](#). Per un elenco dei problemi noti, vedi [Problemi noti](#).

Argomenti

- [Sto cercando di creare un cluster](#)
- [Sto cercando di eseguire un lavoro](#)
- [Tentativo di aggiornamento di un cluster](#)
- [Tentativo di accesso allo storage](#)
- [Tentativo di eliminazione di un cluster](#)
- [Sto cercando di aggiornare lo AWS ParallelCluster stack di API](#)
- [Visualizzazione degli errori nelle inizializzazioni dei nodi di calcolo](#)
- [Risoluzione dei problemi relativi alle metriche di integrità del cluster](#)
- [Risoluzione dei problemi di distribuzione dei cluster](#)
- [Risoluzione dei problemi di distribuzione del cluster utilizzando Terraform](#)
- [Risoluzione dei problemi di scalabilità](#)
- [Gruppi di posizionamento e problemi di avvio delle istanze](#)
- [Sostituzione delle directory](#)
- [Risoluzione dei problemi in Amazon DCV](#)
- [Risoluzione dei problemi nei cluster con integrazione AWS Batch](#)
- [Risoluzione dei problemi di integrazione multiutente con Active Directory](#)
- [Risoluzione dei problemi relativi alle AMI personalizzate](#)
- [Risoluzione dei problemi di un timeout di aggiornamento del cluster quando cfn-hup non è in esecuzione](#)
- [Risoluzione dei problemi di rete](#)
- [Aggiornamento del cluster non riuscito a causa di un'azione personalizzata onNodeUpdated](#)
- [Visualizzazione degli errori con la configurazione personalizzata Slurm](#)

- [Allarmi cluster](#)
- [Risoluzione delle modifiche alla configurazione del sistema operativo che causano errori o guasti](#)

Sto cercando di creare un cluster

Se si utilizza la AWS ParallelCluster versione 3.5.0 e successive per creare un cluster e la creazione di un cluster non è riuscita con `--rollback-on-failure` set to `false`, utilizzare il comando [pcluster describe-cluster](#) CLI per ottenere informazioni sullo stato e sull'errore. In questo caso, l'`pcluster describe-cluster` output previsto `clusterStatus` è `CREATE_FAILED`. Controlla la `failures` sezione dell'output per trovare la `failureCode` e `failureReason`. Quindi, nella sezione seguente, trova la corrispondenza `failureCode` per ulteriori informazioni sulla risoluzione dei problemi. Per ulteriori informazioni, consulta [pcluster describe-cluster](#).

Nelle sezioni seguenti, ti consigliamo di controllare i log sul nodo principale, come i `/var/log/chef-client.log` file `/var/log/cfn-init.log` and. Per ulteriori informazioni sui AWS ParallelCluster log e su come visualizzarli, consulta [Registri chiave per il debug](#) e [Recupero e conservazione dei log](#)

Se non ne hai un `failureCode`, accedi alla CloudFormation console per visualizzare lo stack del cluster. Controlla la `Status Reason` presenza `HeadNodeWaitCondition` o gli errori su altre risorse per trovare ulteriori dettagli sugli errori. Per ulteriori informazioni, consulta [Visualizza CloudFormation gli eventi su CREATE_FAILED](#). Controlla i `/var/log/chef-client.log` file `/var/log/cfn-init.log` and sul nodo principale. Se la creazione del cluster non riesce a causa di un errore di creazione del nodo principale e i log del cluster non sono disponibili nel gruppo di log del cluster, è necessario conservare il cluster in caso di errore, specificare `--rollback-on-failure = True` e recuperare i log dall'interno del nodo principale stesso.

failureCode è OnNodeConfiguredExecutionFailure

- Perché ha fallito?

Hai fornito uno script personalizzato nella sezione `OnNodeConfigured` del nodo principale della configurazione per creare un cluster. Tuttavia, lo script personalizzato non è stato eseguito.

- Come risolvere?

Controlla il `/var/log/cfn-init.log` file per saperne di più sull'errore e su come risolvere il problema nello script personalizzato. Verso la fine di questo registro, potresti visualizzare le

informazioni di esecuzione relative allo `OnNodeConfigured` script dopo il `Running` command `runpostinstall` messaggio.

failureCode è OnNodeConfiguredDownloadFailure

- Perché ha fallito?

Hai fornito uno script personalizzato nella sezione `OnNodeConfigured` del nodo principale della configurazione per creare un cluster. Tuttavia, non è stato possibile scaricare lo script personalizzato.

- Come risolvere?

Assicurati che l'URL sia valido e che l'accesso sia configurato correttamente. Per ulteriori informazioni sulla configurazione degli script di bootstrap personalizzati, consulta [Azioni bootstrap personalizzate](#)

Controllate il file. `/var/log/cfn-init.log` Verso la fine di questo registro, dopo il `Running` command `runpostinstall` messaggio è possibile che vengano visualizzate le informazioni sull'esecuzione relative all'elaborazione degli `OnNodeConfigured` script, incluso il download.

failureCode è OnNodeConfiguredFailure

- Perché ha fallito?

Hai fornito uno script personalizzato nella sezione `OnNodeConfigured` del nodo principale della configurazione per creare un cluster. Tuttavia, l'uso dello script personalizzato non è riuscito nella distribuzione del cluster. Non è possibile determinare una causa immediata e sono necessarie ulteriori indagini.

- Come risolvere?

Controlla il `/var/log/cfn-init.log` file. Verso la fine di questo registro, è possibile che vengano visualizzate le informazioni di esecuzione relative all'elaborazione degli `OnNodeConfigured` script dopo il `Running` command `runpostinstall` messaggio.

failureCode è OnNodeStartExecutionFailure

- Perché ha fallito?

Hai fornito uno script personalizzato nella sezione `OnNodeStart` del nodo principale della configurazione per creare un cluster. Tuttavia, lo script personalizzato non è stato eseguito.

- Come risolvere?

Controlla il `/var/log/cfn-init.log` file per saperne di più sull'errore e su come risolvere il problema nello script personalizzato. Verso la fine di questo registro, potresti visualizzare le informazioni di esecuzione relative allo `OnNodeStart` script dopo il `Running` command `runpreinstall` messaggio.

failureCode è OnNodeStartDownloadFailure

- Perché ha fallito?

Hai fornito uno script personalizzato nella sezione `OnNodeStart` del nodo principale della configurazione per creare un cluster. Tuttavia, non è stato possibile scaricare lo script personalizzato.

- Come risolvere?

Assicurati che l'URL sia valido e che l'accesso sia configurato correttamente. Per ulteriori informazioni sulla configurazione degli script di bootstrap personalizzati, consulta. [Azioni bootstrap personalizzate](#)

Controllate il file. `/var/log/cfn-init.log` Verso la fine di questo registro, dopo il `Running` command `runpreinstall` messaggio è possibile che vengano visualizzate le informazioni sull'esecuzione relative all'elaborazione degli `OnNodeStart` script, incluso il download.

failureCode è OnNodeStartFailure

- Perché ha fallito?

Hai fornito uno script personalizzato nella sezione `OnNodeStart` del nodo principale della configurazione per creare un cluster. Tuttavia, l'uso dello script personalizzato non è riuscito nella distribuzione del cluster. Non è possibile determinare una causa immediata e sono necessarie ulteriori indagini.

- Come risolvere?

Controlla il `/var/log/cfn-init.log` file. Verso la fine di questo registro, è possibile che vengano visualizzate le informazioni di esecuzione relative all'elaborazione degli `OnNodeStart` script dopo il `Running command runpreinstall` messaggio.

failureCode è EbsMountFailure

- Perché ha fallito?

Il volume EBS definito nella configurazione del cluster non è stato montato.

- Come risolvere?

Controlla il `/var/log/chef-client.log` file per i dettagli sull'errore.

failureCode è EfsMountFailure

- Perché ha fallito?

Il volume Amazon EFS definito nella configurazione del cluster non è stato montato.

- Come risolvere?

Se hai definito un file system Amazon EFS esistente, assicurati che il traffico sia consentito tra il cluster e il file system. Per ulteriori informazioni, consulta [SharedStorage/EfsSettings/FileSystemId](#).

Controlla il `/var/log/chef-client.log` file per i dettagli sull'errore.

failureCode è FsxMountFailure

- Perché ha fallito?

Il FSx file system Amazon definito nella configurazione del cluster non è riuscito a montarlo.

- Come risolvere?

Se hai definito un FSx file system Amazon esistente, assicurati che il traffico sia consentito tra il cluster e il file system. Per ulteriori informazioni, consulta [SharedStorage/FsxLustreSettings/FileSystemId](#).

Controlla il `/var/log/chef-client.log` file per i dettagli sull'errore.

failureCode è RaidMountFailure

- Perché ha fallito?

I volumi RAID definiti nella configurazione del cluster non sono stati montati.

- Come risolvere?

Controlla il `/var/log/chef-client.log` file per i dettagli sull'errore.

failureCode è AmiVersionMismatch

- Perché ha fallito?

La AWS ParallelCluster versione utilizzata per creare l'AMI personalizzata è diversa dalla AWS ParallelCluster versione utilizzata per configurare il cluster. Nella CloudFormation console, visualizza i dettagli CloudFormation dello stack del cluster e seleziona la casella Status Reason HeadNodeWaitCondition per ottenere ulteriori dettagli sulle AWS ParallelCluster versioni e sull'AMI. Per ulteriori informazioni, consulta [Visualizza CloudFormation gli eventi su CREATE_FAILED](#).

- Come risolvere?

Assicurati che la AWS ParallelCluster versione utilizzata per creare l'AMI personalizzata sia la stessa AWS ParallelCluster utilizzata per configurare il cluster. Puoi modificare la versione AMI personalizzata o la versione `pcluster` CLI per renderle uguali.

failureCode è InvalidAmi

- Perché ha fallito?

L'AMI personalizzata non è valida perché non è stata creata utilizzando AWS ParallelCluster.

- Come risolvere?

Usa il `pcluster build-image` comando per creare un AMI impostando l'AMI come immagine principale. Per ulteriori informazioni, consulta [pcluster build-image](#).

failureCode è **HeadNodeBootstrapFailure** con **failureReason**

Failed to setup the head node.

- Perché ha fallito?

Non è possibile determinare una causa immediata e sono necessarie ulteriori indagini. Ad esempio, è possibile che il cluster sia in stato protetto e ciò potrebbe essere causato da un mancato provisioning della flotta di elaborazione statica.

- Come risolvere?

Controlla il `/var/log/chef-client.log` file per i dettagli sull'errore.

Note

Se vedi `RuntimeError un'eccezioneCluster state has been set to PROTECTED mode due to failures detected in static node provisioning`, il cluster è in stato protetto. Per ulteriori informazioni, consulta [Come eseguire il debug della modalità protetta](#).

failureCode è scaduto **HeadNodeBootstrapFailure** il timeout per la creazione del **failureReason** cluster.

- Perché ha fallito?

Per impostazione predefinita, è previsto un limite di tempo di 30 minuti per il completamento della creazione del cluster. Se la creazione del cluster non viene completata entro questo lasso di tempo, la creazione del cluster fallisce con un errore di timeout. La creazione del cluster può scadere per diversi motivi. Ad esempio, gli errori di timeout possono essere causati da un errore di creazione del nodo principale, da un problema di rete, da script personalizzati che impiegano troppo tempo per essere eseguiti nel nodo principale, da un errore in uno script personalizzato eseguito nei nodi di calcolo o da lunghi tempi di attesa per il provisioning dei nodi di calcolo. Non è possibile determinare una causa immediata e sono necessarie ulteriori indagini.

- Come risolvere?

Controlla i `/var/log/chef-client.log` file `/var/log/cfn-init.log` and per i dettagli sull'errore. Per ulteriori informazioni sui AWS ParallelCluster log e su come ottenerli, consulta [Registri chiave per il debug](#) e [Recupero e conservazione dei log](#).

Potresti scoprire quanto segue in questi registri.

- Vedendo **Waiting for static fleet capacity provisioning** verso la fine del **chef-client.log**

Ciò indica che la creazione del cluster è scaduta in attesa dell'accensione dei nodi statici. Per ulteriori informazioni, consulta [Visualizzazione degli errori nelle inizializzazioni dei nodi di calcolo](#).

- Lo script di **OnNodeStart** visualizzazione **OnNodeConfigured** o avvio del nodo non è terminato alla fine del **cfn-init.log**

Ciò indica che l'esecuzione dello script `OnNodeConfigured` o dello script `OnNodeStart` personalizzato ha impiegato molto tempo e ha causato un errore di timeout. Verifica che lo script personalizzato non presenti problemi che potrebbero causarne l'esecuzione prolungata. Se lo script personalizzato richiede molto tempo per essere eseguito, valuta la possibilità di modificare il limite di timeout aggiungendo una `DevSettings` sezione al file di configurazione del cluster, come mostrato nell'esempio seguente:

```
DevSettings:
  Timeouts:
    HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- Impossibile trovare i log o il nodo principale non è stato creato correttamente

È possibile che il nodo principale non sia stato creato correttamente e che i log non possano essere trovati. In questo caso, puoi ottenere ulteriori dettagli sugli errori controllando gli eventi CloudFormation dello stack e il registro della console del nodo principale. Puoi recuperare il log della console del nodo principale tramite la console Amazon EC2 o eseguendo il seguente comando CLI di Amazon EC2:

```
aws ec2 get-console-output --instance-id HEAD_NODE_INSTANCE_ID --output text
```

failureCode è **HeadNodeBootstrapFailure** con **failureReason** Failed to bootstrap the head node.

- Perché ha fallito?

Non è possibile determinare una causa immediata e sono necessarie ulteriori indagini.

- Come risolvere?

Controlla i `/var/log/chef-client.log` file `/var/log/cfn-init.log` e.

failureCode è **ResourceCreationFailure**

- Perché ha fallito?

La creazione di alcune risorse non è riuscita durante il processo di creazione del cluster. L'errore può verificarsi per vari motivi. Ad esempio, gli errori di creazione delle risorse possono essere causati da problemi di capacità o da una policy IAM configurata in modo errato.

- Come risolvere?

Nella CloudFormation console, visualizza lo stack del cluster per verificare ulteriori dettagli sugli errori di creazione delle risorse.

failureCode è **ClusterCreationFailure**

- Perché ha fallito?

Non è possibile determinare una causa immediata e sono necessarie ulteriori indagini.

- Come risolvere?

Nella CloudFormation console, visualizza lo stack del cluster e controlla la casella Status Reason per HeadNodeWaitCondition trovare ulteriori dettagli sull'errore.

Controlla i `/var/log/chef-client.log` file `/var/log/cfn-init.log` e.

Vedere **WaitCondition timed out...** in CloudFormation pila

Per ulteriori informazioni, consulta [failureCode è scaduto HeadNodeBootstrapFailure il timeout per la creazione del failureReason cluster..](#)

Vedere **Resource creation cancelled** in pila CloudFormation

Per ulteriori informazioni, consulta [failureCode è ResourceCreationFailure.](#)

Visualizzazione **Failed to run cfn-init...** o altri errori nello stack CloudFormation

Controlla `/var/log/cfn-init.log` e `/var/log/chef-client.log` per ulteriori dettagli sull'errore.

La visione **chef-client.log** finisce con **INFO: Waiting for static fleet capacity provisioning**

Ciò è correlato al timeout di creazione del cluster in attesa dell'accensione dei nodi statici. Per ulteriori informazioni, consulta [Visualizzazione degli errori nelle inizializzazioni dei nodi di calcolo.](#)

Vedendo **Failed to run preinstall or postinstall in cfn-init.log**

Hai uno `OnNodeStart` script `OnNodeConfigured` or nella `HeadNode` sezione di configurazione del cluster. Lo script non funziona correttamente. Controlla il `/var/log/cfn-init.log` file per i dettagli sugli errori degli script personalizzati.

Visualizzazione **This AMI was created with xxx, but is trying to be used with xxx...** in CloudFormation pila

Per ulteriori informazioni, consulta [failureCode è AmiVersionMismatch.](#)

Vedere **This AMI was not baked by AWS ParallelCluster...** in pila CloudFormation

Per ulteriori informazioni, consulta [failureCode è InvalidAmi.](#)

Il `pcluster create-cluster` comando Seeing non viene eseguito localmente

~/parallelcluster/pcluster-cli.logControllate il file system locale per i dettagli sull'errore.

Supporto aggiuntivo

Segui le istruzioni per la risoluzione dei problemi riportate in [Risoluzione dei problemi di distribuzione dei cluster](#).

Verifica se il tuo scenario è coperto nella sezione [Problemi GitHub noti](#) AWS ParallelCluster all'indirizzo GitHub.

Sto cercando di eseguire un lavoro

La sezione seguente fornisce possibili soluzioni per la risoluzione dei problemi in caso di problemi durante il tentativo di eseguire un processo.

`srun` processo interattivo fallisce e restituisce un errore `srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf`

- Perché ha fallito?

Hai eseguito il `srun` comando per inviare un lavoro, quindi hai aumentato la dimensione di una coda utilizzando il `pcluster update-cluster` comando senza riavviare i Slurm demoni dopo il completamento dell'aggiornamento.

Slurm organizza i Slurm demoni in una gerarchia ad albero per ottimizzare la comunicazione. Questa gerarchia viene aggiornata solo all'avvio dei demoni.

Si supponga `srun` di dover avviare un processo e quindi eseguire il `pcluster update-cluster` comando per aumentare le dimensioni della coda. I nuovi nodi di calcolo vengono avviati come parte dell'aggiornamento. Quindi, mette in Slurm coda il lavoro su uno dei nuovi nodi di calcolo. In questo caso, sia i Slurm demoni che quelli `srun` non rilevano i nuovi nodi di calcolo. `srun` restituisce un errore perché non rileva i nuovi nodi.

- Come risolvere?

Riavvia i Slurm daemon su tutti i nodi di calcolo, quindi usali `srun` per inviare il lavoro. È possibile pianificare il riavvio dei Slurm demoni eseguendo il `scontrol reboot` comando che riavvia i nodi di calcolo. Per ulteriori informazioni, consulta [scontrol reboot](#) nella documentazione. Slurm Puoi anche riavviare manualmente Slurm i demoni sui nodi di calcolo richiedendo il riavvio dei servizi corrispondenti. `systemd`

Job è bloccato nello CF stato con **squeue** il comando

Questo potrebbe essere un problema con l'accensione dei nodi dinamici. Per ulteriori informazioni, consulta [Visualizzazione degli errori nelle inizializzazioni dei nodi di calcolo](#).

Esecuzione di lavori su larga scala e visualizzazione **nfsd: too many open connections, consider increasing the number of threads in /var/log/messages**

Con un file system in rete, quando vengono raggiunti i limiti di rete, aumentano anche i tempi di I/O attesa. Ciò può comportare blocchi automatici perché la rete viene utilizzata per scrivere dati sia per la rete che per le metriche. I/O

Nelle istanze di quinta generazione, utilizziamo il driver ENA per esporre i contatori di pacchetti. Questi contatori contano i pacchetti formati dal AWS momento in cui la rete raggiunge i limiti di larghezza di banda delle istanze. Puoi controllare questi contatori per vedere se sono maggiori di 0. Se lo sono, significa che hai superato i limiti di larghezza di banda. È possibile visualizzare questi contatori eseguendo. `ethtool -S eth0 | grep exceeded`

Il superamento dei limiti di rete è spesso il risultato del supporto di troppe connessioni NFS. Questa è una delle prime cose da verificare quando si raggiungono o superano i limiti di rete.

Ad esempio, l'output seguente mostra i pacchetti eliminati:

```
$ ethtool -S eth0 | grep exceeded
bw_in_allowance_exceeded: 38750610
bw_out_allowance_exceeded: 1165693
pps_allowance_exceeded: 103
conntrack_allowance_exceeded: 0
linklocal_allowance_exceeded: 0
```

Per evitare di ricevere questo messaggio, valuta la possibilità di modificare il tipo di istanza del nodo principale con un tipo di istanza più performante. Prendi in considerazione la possibilità di spostare lo storage dei dati su file system di storage condivisi che non vengono esportati come condivisione NFS, come Amazon EFS o Amazon FSx. Per ulteriori informazioni, consulta [Archiviazione condivisa](#) e consulta le [Best Practice](#) nel AWS ParallelCluster Wiki su GitHub.

Esecuzione di un job MPI

Attivazione della modalità di debug

Per abilitare la modalità di debug OpenMPI, [consultate Quali controlli dispone Open MPI](#) per facilitare il debug.

[Per abilitare la modalità di debug IntelMPI, vedete Altre variabili di ambiente.](#)

Visualizzazione **MPI_ERRORS_ARE_FATAL** e **OPAL ERROR** inserimento dei risultati del lavoro

Questi codici di errore provengono dal livello MPI dell'applicazione. Per informazioni su come ottenere i registri di debug MPI dall'applicazione, consulta [Attivazione della modalità di debug](#).

Una possibile causa di questo errore è che l'applicazione è stata compilata per un'implementazione MPI specifica, ad esempio OpenMPI, e si sta tentando di eseguirla con un'implementazione MPI diversa, ad esempio IntelMPI. Assicurati di compilare ed eseguire l'applicazione con la stessa implementazione MPI.

Utilizzo **mpirun** con DNS gestito disabilitato

Per i cluster creati con [SlurmSettings/Dns/DisableManagedDns](#) e [UseEc2Hostnames](#) impostati su `true`, il nome del Slurm nodo non viene risolto dal DNS. Slurm può avviare i processi MPI quando `nodenames` non sono abilitati e se il job MPI viene eseguito in un contesto. Slurm Si consiglia di seguire le indicazioni contenute nella Guida per [l'utente di Slurm MPI per eseguire i job MPI](#) con Slurm.

Tentativo di aggiornamento di un cluster

La sezione seguente fornisce possibili soluzioni per la risoluzione dei problemi che potrebbero verificarsi durante l'aggiornamento di un cluster.

pcluster update-cluster il comando non viene eseguito localmente

~/parallelcluster/pcluster-cli.log Controllate il file system locale per i dettagli sull'errore.

Vedere **clusterStatus** avviene **UPDATE_FAILED** tramite **pcluster describe-cluster** comando

Causa principale

Per identificare la causa principale dell'errore, il punto di partenza è esaminare gli eventi dello stack del cluster e `/var/log/chef-client.log` il nodo principale.

Una possibile causa è che almeno un nodo del cluster non ha applicato l'aggiornamento. È possibile recuperare l'elenco dei nodi che non sono stati aggiornati `/var/log/chef-client.log` nel nodo principale effettuando una ricerca `Check cluster readiness` nel registro.

Verifica se il problema è menzionato nella sezione [Problemi GitHub noti](#) all'indirizzo AWS ParallelCluster on GitHub.

Prevenire

Un aggiornamento del cluster può avere esito negativo se almeno un nodo del cluster non ha applicato correttamente l'aggiornamento. Per ridurre il rischio di errori di aggiornamento del cluster, si consiglia di chiudere i nodi danneggiati prima di iniziare l'aggiornamento. Un esempio di nodi che potrebbero essere danneggiati sono i nodi di elaborazione bloccati nello `COMPLETING` stato per un periodo più lungo della durata prevista dell'epilogo. Per rilevare tali nodi, puoi eseguire il comando seguente, adattando il `threshold` valore alle tue esigenze (il valore deve essere maggiore della durata massima prevista per gli epiloghi).

```
$ scontrol show nodes --json | jq -r --argjson threshold 60 '
  .nodes[] | select(.state | index("COMPLETING")) |
  select((now - .last_busy.number) > $threshold) |
  .name
'
```

Recupero

Se l'aggiornamento non è riuscito, il rollback è il meccanismo previsto per ripristinare lo stato del cluster.

Se il rollback non è riuscito, lo stato del cluster non è deterministico. In questo caso, è possibile che sia `clustermgtd` stato interrotto per impedire l'amplificazione dei guasti. Si consiglia di avviarlo eseguendo il seguente comando sul nodo principale. Adatta la versione di Python a quella fornita con la tua versione: AWS ParallelCluster

```
$ /opt/parallelcluster/pyenv/versions/3.12.11/envs/cookbook_virtualenv/bin/supervisorctl start clustermgtd
```

L'aggiornamento del cluster è scaduto

Potrebbe trattarsi di un problema correlato alla `cfn-hup` mancata esecuzione. Se il `cfn-hup` demone viene eliminato da una causa esterna, non viene riavviato automaticamente. Se `cfn-hup` non è in esecuzione, durante un aggiornamento del cluster, lo CloudFormation stack avvia il processo di aggiornamento come previsto, ma la procedura di aggiornamento non viene attivata sul nodo principale e la distribuzione dello stack alla fine scade. Per ulteriori informazioni, consulta Risolvere [Risoluzione dei problemi di un timeout di aggiornamento del cluster quando cfn-hup non è in esecuzione](#) il problema e risolvere il problema.

Tentativo di accesso allo storage

Scopri i suggerimenti per la risoluzione dei problemi relativi al tentativo di accesso allo storage.

Utilizzo di un file system Amazon FSx for Lustre esterno

Assicurati che il traffico sia consentito tra il cluster e il file system. Il file system deve essere associato a un gruppo di sicurezza che consenta il traffico TCP in entrata e in uscita attraverso le porte 988, 1021, 1022 e 1023. Per ulteriori informazioni su come configurare i gruppi di sicurezza, vedere.

[FileSystemId](#)

Utilizzo di un file system Amazon Elastic File System esterno

Assicurati che il traffico sia consentito tra il cluster e il file system. Il file system deve essere associato a un gruppo di sicurezza che consenta il traffico TCP in entrata e in uscita attraverso le porte 988, 1021, 1022 e 1023. Per ulteriori informazioni su come configurare i gruppi di sicurezza, vedere.

[FileSystemId](#)

Tentativo di eliminazione di un cluster

Se si verifica un errore durante il tentativo di eliminare un cluster, nelle sezioni seguenti vengono forniti suggerimenti per la risoluzione dei problemi relativi agli scenari più comuni.

Il **pcluster delete-cluster** comando non viene eseguito localmente

Controllate il `~/.parallelcluster/pcluster-cli.log` file nel file system locale.

Lo stack del cluster non viene eliminato

Se lo stack del cluster non riesce a eliminare, controlla il messaggio relativo agli eventi dello CloudFormation stack.

Controlla se il tuo problema è menzionato nella sezione [Problemi GitHub noti all'indirizzo AWS ParallelCluster on](#). GitHub

Sto cercando di aggiornare lo AWS ParallelCluster stack di API

Se si verifica un errore, ad esempio `UPDATE_FAILED` quando si tenta di aggiornare lo stack di AWS ParallelCluster API, si consiglia di cercare una soluzione nelle sezioni Problemi noti del [AWS ParallelCluster Wiki](#) su GitHub. Ad esempio, consulta [ParallelCluster API Stack Upgrade Fails for ECR resources](#), che identifica un possibile problema e fornisce opzioni di mitigazione.

Visualizzazione degli errori nelle inizializzazioni dei nodi di calcolo

Le sezioni seguenti forniscono suggerimenti per la risoluzione dei problemi in caso di errori nelle inizializzazioni dei nodi di calcolo. Ciò include gli errori di bootstrap, la visualizzazione degli errori nei log e gli indirizzi da seguire se nessuno degli scenari si adatta alla situazione specifica.

Argomenti

- [Vedere in Node bootstrap error clustermgtd.log](#)
- [Ho configurato le prenotazioni di capacità su richiesta \(ODCRs\) o le istanze riservate zonali](#)
- [An error occurred \(VcpuLimitExceeded\)Attivo slurm_resume.log quando non riesco a eseguire un processo o attivo quando non riesco a creare un cluster clustermgtd.log](#)
- [An error occurred \(InsufficientInstanceCapacity\)Attivo slurm_resume.log quando non riesco a eseguire un processo o attivo quando non riesco a creare un cluster clustermgtd.log](#)

- [I nodi di visualizzazione sono DOWN in stato con Reason \(Code:InsufficientInstanceCapacity\)...](#)
- [Vedere dentro cannot change locale \(en_US.utf-8\) because it has an invalid name slurm_resume.log](#)
- [Nessuno degli scenari precedenti si applica alla mia situazione](#)

Vedere in **Node bootstrap error clustermgtd.log**

Il problema è legato al mancato avvio dei nodi di calcolo. Per informazioni su come eseguire il debug di un problema relativo alla modalità protetta del cluster, vedere. [Come eseguire il debug della modalità protetta](#)

Ho configurato le prenotazioni di capacità su richiesta (ODCRs) o le istanze riservate zonali

ODCRs che includono istanze con più interfacce di rete, come P4d, P4de e Trainium (Trn) AWS

Nel file di configurazione del cluster, verifica che si trovi in una sottorete pubblica e che i nodi di elaborazione HeadNode si trovino in una sottorete privata.

ODCRs sono ODCRS mirati

Vedo **Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'**. anche se l'ho già fatto seguendo le istruzioni riportate / **opt/slurm/etc/pcluster/run_instances_overrides.json** in [Avvio di istanze con On-Demand Capacity Reservations \(ODCR\)](#)

Se utilizzi le AWS ParallelCluster versioni da 3.1.1 a 3.2.1 con targeted ODCRs e utilizzi anche il file JSON [run instances override](#), è possibile che il file JSON non sia formattato correttamente. Potresti visualizzare un errore, come il seguente: clustermgtd.log

```
Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'.  
Using default: {} in /var/log/parallelcluster/clustermgtd.
```

Verifica che il formato di file JSON sia corretto eseguendo quanto segue:

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

Verifica **Found RunInstances parameters override. clustermgtd.log** quando la creazione del cluster non è riuscita o **slurm_resume.log** quando l'esecuzione del processo non è riuscita

Se utilizzi il [file run instances override JSON](#), verifica di aver impostato correttamente il nome della coda e il nome delle risorse di calcolo nel file. `/opt/slurm/etc/pcluster/run_instances_overrides.json`

An error occurred (InsufficientInstanceCapacity)Accetto **slurm_resume.log** quando non riesco a eseguire un processo o **clustermgtd.log** quando non riesco a creare un cluster

Utilizzo di PG-ODCR (Placement Group ODCR)

Quando si crea un ODCR con un gruppo di posizionamento associato, è necessario utilizzare lo stesso nome del gruppo di posizionamento nel file di configurazione. Imposta il [nome del gruppo di posizionamento](#) corrispondente nella configurazione del cluster.

Utilizzo di istanze riservate zonali

Se utilizzi istanze riservate zonali con `PlacementGroup/Enabled` to `true` nella configurazione del cluster, potresti visualizzare un errore, come il seguente:

```
We currently do not have sufficient trn1.32xlarge capacity in the Availability Zone
you requested (us-east-1d). Our system will be working on provisioning additional
capacity.
You can currently get trn1.32xlarge capacity by not specifying an Availability Zone in
your request or choosing us-east-1a, us-east-1b, us-east-1c, us-east-1e, us-east-1f.
```

È possibile che ciò si verifichi perché le istanze riservate zonali non sono collocate nella stessa UC (o spine), il che può causare errori di capacità insufficiente (ICEs) quando si utilizzano i gruppi di collocamento. Puoi verificare questo caso disabilitando l'impostazione `PlacementGroup Group` nella configurazione del cluster per determinare se il cluster può allocare le istanze.

An error occurred (VcpuLimitExceeded)Attivo **slurm_resume.log** quando non riesco a eseguire un processo o attivo **clustermgtd.log** quando non riesco a creare un cluster

Verifica i limiti di vCPU sul tuo account per il tipo di istanza Amazon EC2 specifico che stai utilizzando. Se il valore `v` è pari o inferiore CPUs a quello richiesto, richiedi un aumento dei limiti. Per

informazioni su come visualizzare i limiti attuali e richiederne di nuovi, consulta le [quote dei servizi Amazon EC2](#) nella Amazon EC2 User Guide.

An error occurred (InsufficientInstanceCapacity)Attivo **slurm_resume.log** quando non riesco a eseguire un processo o attivo quando non riesco a creare un cluster **clustermgtd.log**

Stai riscontrando un problema di capacità insufficiente. Segui <https://aws.amazon.com/premiumsupport/knowledge-center/ec2> -/per risolvere il problema. insufficient-capacity-errors

I nodi di visualizzazione sono **DOWN** in stato con **Reason (Code:InsufficientInstanceCapacity)...**

Stai riscontrando un problema di capacità insufficiente. Segui <https://aws.amazon.com/premiumsupport/knowledge-center/ec2> -/per risolvere il problema. insufficient-capacity-errors
Per ulteriori informazioni sulla modalità di failover veloce con capacità insufficiente, vedere AWS ParallelCluster. [Slurmfailover rapido con capacità insufficiente del cluster](#)

Vedere dentro **cannot change locale (en_US.utf-8) because it has an invalid name slurm_resume.log**

Ciò può verificarsi se il processo di yum installazione non è riuscito e le impostazioni locali sono state mantenute in uno stato incoerente. Ad esempio, ciò può verificarsi quando un utente termina il processo di installazione.

Per verificare la causa, intraprendi le seguenti azioni:

- Esegui su `pcluster-admin`.

La shell mostra un errore, ad esempiocannot change locale...no such file or directory.

- Esegui `localedef --list`.

Restituisce una lista vuota o non contiene la localizzazione predefinita.

- Controlla l'ultimo yum comando con `yum history` and `yum history info #ID`. L'ultimo ID haReturn-Code: Success?

Se l'ultimo ID non è presente `Return-Code: Success`, gli script di post-installazione potrebbero non essere stati eseguiti correttamente.

Per risolvere il problema, prova a ricostruire la versione locale con `yum reinstall glibc-all-langpacks`. Dopo la ricostruzione, su `pccluster-admin` non mostra alcun errore o avviso se il problema è stato risolto.

Nessuno degli scenari precedenti si applica alla mia situazione

Per risolvere i problemi di inizializzazione dei nodi di calcolo, consulta [Risoluzione dei problemi di inizializzazione dei nodi](#)

Verifica se il tuo scenario è incluso nella sezione Problemi [GitHub noti](#) all'indirizzo `on. AWS ParallelCluster GitHub`

Risoluzione dei problemi relativi alle metriche di integrità del cluster

Le metriche sullo stato del cluster vengono aggiunte alla CloudWatch dashboard di AWS ParallelCluster Amazon a partire dalla AWS ParallelCluster versione 3.6.0. Nelle sezioni seguenti, puoi conoscere i parametri di salute della dashboard e le azioni che puoi intraprendere per risolvere e risolvere i problemi.

Argomenti

- [Visualizzazione del grafico degli errori di provisioning delle istanze](#)
- [Visualizzazione del grafico Unhealthy Instance Errors](#)
- [Visualizzazione del grafico Compute Fleet Idle Time](#)

Visualizzazione del grafico degli errori di provisioning delle istanze

Se vedi un valore diverso da zero nel `Instance Provisioning Errors` grafico, significa che l'istanza Amazon EC2 per il backup dei nodi slurm non è stata avviata sull'API `or. CreateFleet RunInstance`

Vedendo **IAMPolicyErrors**

- Cosa è successo?

Alcune istanze non sono state avviate, a causa di autorizzazioni insufficienti con codice di errore. `UnauthorizedOperation`

- Come risolvere?

Se hai configurato un [InstanceRole](#) o personalizzato [InstanceProfile](#), controlla le tue policy IAM e verifica di utilizzare le credenziali corrette.

Controlla il `clustermgtd` file per i dettagli sugli errori statici dei nodi. Controlla il `slurm_resume.log` file per i dettagli degli errori dinamici del nodo. Usa i dettagli per saperne di più sulle autorizzazioni mancanti che devono essere aggiunte.

Vedendo **VcpuLimitErrors**

- Cosa è successo?

AWS ParallelCluster non è riuscito ad avviare le istanze perché ha raggiunto il limite di vCPU per uno specifico Account AWS tipo di istanza Amazon EC2 configurato per i nodi di calcolo del cluster.

- Come risolvere?

Controlla l'`VcpuLimitExceeded` errore nel `clustermgtd` file per i nodi statici e controlla il `slurm_resume.log` file per i nodi dinamici per ottenere ulteriori dettagli. Per risolvere questo problema, puoi richiedere un aumento dei limiti di vCPU. Per ulteriori informazioni su come visualizzare i limiti attuali e richiederne di nuovi, consulta le [quote dei servizi Amazon Elastic Compute Cloud](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Vedendo **VolumeLimitErrors**

- Cosa è successo?

Hai raggiunto il limite di volume di Amazon EBS sul tuo Account AWS e AWS ParallelCluster non riesci ad avviare istanze con codice `InsufficientVolumeCapacity` di errore o. `VolumeLimitExceeded`

- Come risolvere?

Controlla il `clustermgtd` file per i nodi statici e controlla il `slurm_resume.log` file per i nodi dinamici per ottenere ulteriori dettagli sui limiti di volume. Per risolvere questo problema, puoi

utilizzare un altro Regione AWS, pulire i volumi esistenti o contattare il AWS Support Center per inviare una richiesta di aumento del limite di volume di Amazon EBS.

Vedendo **InsufficientCapacityErrors**

- Cosa è successo?

AWS ParallelCluster non dispone di una capacità sufficiente per avviare istanze Amazon EC2 sui backnode.

- Come risolvere?

Controlla il `clustermgtd` file per i nodi statici e controlla il `slurm_resume.log` file per i nodi dinamici per ottenere dettagli sugli errori di capacità insufficienti. Per risolvere il problema, segui le istruzioni disponibili su <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors>.

OtherInstanceLaunchFailures

- Cosa è successo?

L'istanza Amazon EC2 per il backup dei nodi di calcolo non è stata avviata con l'API `or.CreateFleet` o `RunInstance`.

- Come risolvere?

Controlla il `clustermgtd` file per i nodi statici e controlla il `slurm_resume.log` file per i nodi dinamici per ottenere i dettagli dell'errore.

Visualizzazione del grafico Unhealthy Instance Errors

- Cosa è successo?

Sono state lanciate diverse istanze di calcolo, ma in seguito sono state chiuse perché non funzionanti.

- Come risolvere?

Per ulteriori informazioni sulla risoluzione dei problemi relativi ai nodi non integri, vedere [Risoluzione dei problemi di sostituzioni e terminazioni impreviste dei nodi](#).

Vedendo **InstanceBootstrapTimeoutError**

- Cosa è successo?

Un'istanza non può entrare a far parte del cluster all'interno di `resume_timeout` (per nodi dinamici) o `node_replacement_timeout` (per nodi statici). Ciò può verificarsi se la rete non è configurata correttamente per i nodi di calcolo oppure se gli script personalizzati in esecuzione sul nodo di calcolo impiegano troppo tempo per essere completati.

- Come risolvere?

Per i nodi dinamici, controlla nel `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) l'indirizzo IP del nodo di calcolo e gli errori come i seguenti:

```
Node bootstrap error: Resume timeout expires for node
```

Per i nodi statici, controllate `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) per l'indirizzo IP del nodo di calcolo ed errori come i seguenti:

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

Per ulteriori dettagli, controllate la presenza di errori nel `/var/log/cloud-init-output.log` file. È possibile recuperare gli indirizzi IP dei nodi di calcolo problematici dai file di registro `clustermgtd` `eslurm_resume`.

Vedendo **EC2HealthCheckErrors**

- Cosa è successo?

Un'istanza non ha superato il controllo dello stato di Amazon EC2.

- Come risolvere?

Per informazioni su come risolvere questo problema, consulta Risolvere i [problemi relativi alle istanze con controlli](#) di stato non riusciti.

Vedendo **ScheduledEventHealthCheckErrors**

- Cosa è successo?

Un'istanza non ha superato il controllo dello stato di un evento pianificato da Amazon EC2 e non è integra.

- Come risolvere?

Per informazioni su come risolvere questo problema, consulta [Eventi pianificati per le tue](#) istanze.

Vedendo **NoCorrespondingInstanceErrors**

- Cosa è successo?

AWS ParallelCluster non riesce a trovare istanze che supportano i nodi. Probabilmente i nodi si sono interrotti automaticamente durante le operazioni di bootstrap.

[SlurmQueues/CustomActions/OnNodeStart](#) possono produrre errori di [OnNodeConfigured](#) script o di rete. **NoCorrespondingInstanceErrors**

- Come risolvere?

Per ulteriori dettagli, consulta la sezione `/var/log/cloud-init-output.log` dedicata al nodo di calcolo.

Visualizzazione del grafico Compute Fleet Idle Time

Visualizzazione di un valore **MaxDynamicNodeIdleTime** significativamente più lungo della soglia di scalabilità del tempo di inattività

- Cosa è successo?

La tua istanza non si sta concludendo correttamente. **MaxDynamicNodeIdleTime** mostra il tempo massimo, in secondi, di inattività di un nodo dinamico, supportato da un'istanza Amazon EC2. La soglia di Idle Time Scaledown è derivata dal parametro di configurazione del cluster.

[ScaledownIdleTime](#) Quando un nodo di calcolo è inattivo per più di secondi di Idle Time Scaledown, spegne il nodo e termina l'istanza di Slurm backup. AWS ParallelCluster In questo caso, qualcosa impedisce la chiusura dell'istanza.

- Come risolvere?

Per ulteriori informazioni su questo problema, vedere [Sostituzione, interruzione o spegnimento di istanze e nodi problematici](#) in [Risoluzione dei problemi di scalabilità](#).

Risoluzione dei problemi di distribuzione dei cluster

Se il cluster non viene creato e ripristina la creazione dello stack, puoi esaminare i file di registro per diagnosticare il problema. Il messaggio di errore è probabilmente simile al seguente output:

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
--cluster-configuration cluster-config.yaml
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.15.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}

$ pcluster describe-cluster --cluster-name mycluster --region eu-west-1
{
  "creationTime": "2021-09-06T11:03:47.696Z",
  ...
  "cloudFormationStackStatus": "ROLLBACK_IN_PROGRESS",
  "clusterName": "mycluster",
  "computeFleetStatus": "UNKNOWN",
  "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
  "lastUpdatedTime": "2021-09-06T11:03:47.696Z",
  "region": "eu-west-1",
  "clusterStatus": "CREATE_FAILED"
}
```

Argomenti

- [Visualizza CloudFormation gli eventi su CREATE_FAILED](#)
- [Usa la CLI per visualizzare i flussi di log](#)
- [Ricrea il cluster fallito con rollback-on-failure](#)

Visualizza CloudFormation gli eventi su **CREATE_FAILED**

È possibile utilizzare la console o la AWS ParallelCluster CLI per visualizzare CloudFormation gli eventi CREATE_FAILED relativi agli errori per individuare la causa principale.

Argomenti

- [Visualizza gli eventi nella console CloudFormation](#)
- [Usa la CLI per visualizzare e CloudFormation filtrare gli eventi su CREATE_FAILED](#)

Visualizza gli eventi nella console CloudFormation

Per visualizzare ulteriori informazioni sulla causa dello "CREATE_FAILED" stato, puoi utilizzare la CloudFormation console.

Visualizza i messaggi di CloudFormation errore dalla console.

1. Accedi a Console di gestione AWS e vai a <https://console.aws.amazon.com/cloudformation>.
2. Selezionare la pila denominata *cluster_name*.
3. Scegli la scheda Eventi.
4. Controlla lo stato della risorsa che non è stata creata scorrendo l'elenco degli eventi delle risorse per ID logico. Se la creazione di una sottoattività non è riuscita, procedi a ritroso per trovare l'evento relativo alla risorsa non riuscita.
5. Ad esempio, se viene visualizzato il seguente messaggio di stato, è necessario utilizzare tipi di istanza che non superino il limite di vCPU corrente o che non richiedano una maggiore capacità di vCPU.

```
2022-02-04 16:09:44 UTC-0800 HeadNode CREATE_FAILED You have requested more vCPU
capacity than your current vCPU limit of 0 allows
    for the instance bucket that the specified instance type belongs to. Please
visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to
this limit.
    (Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request
ID: a9876543-b321-c765-d432-dcba98766789; Proxy: null).
```

Usa la CLI per visualizzare e CloudFormation filtrare gli eventi su **CREATE_FAILED**

Per diagnosticare il problema di creazione del cluster, puoi utilizzare il [pcluster get-cluster-stack-events](#) comando filtrando in base allo stato. CREATE_FAILED Per ulteriori informazioni, consulta [Filtraggio dell' AWS CLI output nella Guida per l'AWS Command Line Interface utente](#).

```
$ pcluster get-cluster-stack-events --cluster-name mycluster --region eu-west-1 \
  --query 'events[?resourceStatus==`CREATE_FAILED`]'
[
  {
    "eventId": "3ccdedd0-0f03-11ec-8c06-02c352fe2ef9",
    "physicalResourceId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "The following resource(s) failed to create: [HeadNode].",
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "mycluster",
    "resourceType": "AWS::CloudFormation::Stack",
    "timestamp": "2021-09-06T11:11:51.780Z"
  },
  {
    "eventId": "HeadNode-CREATE_FAILED-2021-09-06T11:11:50.127Z",
    "physicalResourceId": "i-04e91cc1f4ea796fe",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "Received FAILURE signal with UniqueId
i-04e91cc1f4ea796fe",
    "resourceProperties": "{\"LaunchTemplate\":{\"Version\":\"1\",\"LaunchTemplateId
\":\"lt-057d2b1e687f05a62\"}}",
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "HeadNode",
    "resourceType": "AWS::EC2::Instance",
    "timestamp": "2021-09-06T11:11:50.127Z"
  }
]
```

Nell'esempio precedente, l'errore riguardava la configurazione del nodo principale.

Usa la CLI per visualizzare i flussi di log

Per eseguire il debug di questo tipo di problema, puoi elencare i flussi di log disponibili dal nodo principale filtrando `node-type` e quindi [pcluster list-cluster-log-streams](#) analizzando il contenuto dei flussi di log.

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
--filters 'Name=node-type,Values=HeadNode'
{
  "logStreams": [
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      ...
    },
    ...
  ]
}
```

I due flussi di log principali che è possibile utilizzare per trovare gli errori di inizializzazione sono i seguenti:

- `cfn-init` è il registro dello script. `cfn-init` Per prima cosa controlla questo flusso di log. È probabile che tu veda l'Command `chef` failed errore in questo registro. Guarda le righe immediatamente precedenti a questa riga per ulteriori dettagli relativi al messaggio di errore. Per ulteriori informazioni, vedere [cfn-init](#).
- `cloud-init` [è il log per cloud-init](#). Se non vedi nulla in `cfn-init`, prova a controllare successivamente questo registro.

Puoi recuperare il contenuto del flusso di log utilizzando [pcluster get-cluster-log-events](#) (nota l'`--limit 5` opzione per limitare il numero di eventi recuperati):

```
$ pcluster get-cluster-log-events --cluster-name mycluster \
  --region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init \
  --limit 5
{
  "nextToken": "f/36370880979637159565202782352491087067973952362220945409/s",
  "prevToken": "b/36370880752972385367337528725601470541902663176996585497/s",
  "events": [
    {
      "message": "2021-09-06 11:11:39,049 [ERROR] Unhandled exception during build:
Command runpostinstall failed",
      "timestamp": "2021-09-06T11:11:39.049Z"
    },
    {
      "message": "Traceback (most recent call last):\n  File \"/opt/aws/bin/
cfn-init\", line 176, in <module>\n    worklog.build(metadata, configSets)\n
  File \"/usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line
  135, in build\n    Contractor(metadata).build(configSets, self)\n  File \"/
usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 561, in
  build\n    self.run_config(config, worklog)\n  File \"/usr/lib/python3.7/
site-packages/cfnbootstrap/construction.py\", line 573, in run_config\n
  CloudFormationCarpenter(config, self._auth_config).build(worklog)\n  File \"/usr/
lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 273, in build\n
  self._config.commands)\n  File \"/usr/lib/python3.7/site-packages/cfnbootstrap/
command_tool.py\", line 127, in apply\n    raise ToolError(u\"Command %s failed\" %
  name)\",
      "timestamp": "2021-09-06T11:11:39.049Z"
    },
    {
      "message": "cfnbootstrap.construction_errors.ToolError: Command runpostinstall
failed",
      "timestamp": "2021-09-06T11:11:39.049Z"
    },
    {
      "message": "2021-09-06 11:11:49,212 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.eu-west-1.amazonaws.com",
      "timestamp": "2021-09-06T11:11:49.212Z"
    },
    {
      "message": "2021-09-06 11:11:49,213 [DEBUG] Signaling resource HeadNode in stack
mycluster with unique ID i-04e91cc1f4ea796fe and status FAILURE",
```

```

    "timestamp": "2021-09-06T11:11:49.213Z"
  }
]
}

```

Nell'esempio precedente, l'errore è causato da un `runpostinstall` errore, quindi è strettamente correlato al contenuto dello script di bootstrap personalizzato utilizzato nel parametro di `OnNodeConfigured` configurazione di. [CustomActions](#)

Ricrea il cluster fallito con **rollback-on-failure**

AWS ParallelCluster crea flussi di CloudWatch log del cluster in gruppi di log. È possibile visualizzare questi registri nei dashboard personalizzati o nei CloudWatch gruppi di log della console. Per ulteriori informazioni, consultare [Integrazione con Amazon CloudWatch Logs](#) e [CloudWatch Pannello di controllo Amazon](#). Se non sono disponibili flussi di log, l'errore potrebbe essere causato dallo script di bootstrap [CustomActions](#) personalizzato o da un problema relativo all'AMI. Per diagnosticare il problema di creazione in questo caso, crea nuovamente il cluster utilizzando [pcluster create-cluster](#), incluso il parametro impostato su. `--rollback-on-failure false` Quindi, utilizzate SSH per visualizzare il cluster, come illustrato di seguito:

```

$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
  --cluster-configuration cluster-config.yaml --rollback-on-failure false
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.15.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
$ pcluster ssh --cluster-name mycluster

```

Dopo aver effettuato l'accesso al nodo principale, dovresti trovare tre file di log principali che puoi usare per trovare l'errore.

- `/var/log/cfn-init.log` è il registro dello `cfn-init` script. Per prima cosa controlla questo registro. È probabile che venga visualizzato un errore come `Command chef failed` in questo

registro. Guarda le righe immediatamente precedenti a questa riga per ulteriori dettagli relativi al messaggio di errore. Per ulteriori informazioni, vedere [cfn-init](#).

- `/var/log/cloud-init.log` è il log per `cloud-init`. Se non vedi `nullacfn-init.log`, prova a controllare successivamente questo registro.
- `/var/log/cloud-init-output.log` è l'output dei comandi eseguiti da `cloud-init`. Questo include l'output di `cfn-init`. Nella maggior parte dei casi, non è necessario consultare questo registro per risolvere questo tipo di problema.

Risoluzione dei problemi di distribuzione del cluster utilizzando Terraform

Questa sezione è rilevante per i cluster che sono stati distribuiti utilizzando Terraform.

ParallelCluster API non trovata

La pianificazione potrebbe fallire perché l'ParallelCluster API non può essere trovata. In questo caso, l'errore restituito sarebbe qualcosa del tipo:

```
Planning failed. Terraform encountered an error while generating this plan.

#
# Error: Unable to retrieve ParallelCluster API cloudformation stack.
#
#   with provider["registry.terraform.io/aws-tf/aws-parallelcluster"],
#   on providers.tf line 6, in provider "aws-parallelcluster":
#     6: provider "aws-parallelcluster" {
#
# operation error CloudFormation: DescribeStacks, https response error StatusCode: 400,
# RequestID: REQUEST_ID, api error ValidationError: Stack with id PCAPI_STACK_NAME does
# not exist
```

Per risolvere questo errore, distribuisce l'ParallelCluster API nell'account in cui verranno creati i cluster. Per informazioni, consulta [the section called “Creare un cluster con Terraform”](#).

Utente non autorizzato a chiamare l'API ParallelCluster

La pianificazione potrebbe fallire perché l'IAM role/user che hai ipotizzato per implementare il tuo progetto Terraform non dispone delle autorizzazioni per interagire con l'API. ParallelCluster In questo caso, l'errore restituito sarebbe qualcosa del tipo:

```
Planning failed. Terraform encountered an error while generating this plan.
```

```
# Error: 403 Forbidden
#
#   with
#     module.parallelcluster_clusters.module.clusters[0].pcluster_cluster.managed_configs["DemoCluster"],
#   on .terraform/modules/parallelcluster_clusters/modules/clusters/main.tf line 35, in
#   resource "pcluster_cluster" "managed_configs":
#     35: resource "pcluster_cluster" "managed_configs" {
#
#   [{"Message": "User: USER_ARN is not authorized to perform: execute-api:Invoke on
#     resource: PC_API_REST_RESOURCE with an explicit deny"}]
# }
```

Per risolvere questo errore, configura il ParallelCluster provider in modo che utilizzi il ruolo ParallelCluster API per interagire con l'API.

```
provider "aws-parallelcluster" {
  region      = var.region
  profile      = var.profile
  api_stack_name = var.api_stack_name
  **use_user_role** **= true**
}
```

Risoluzione dei problemi di scalabilità

Questa sezione è rilevante per i cluster che sono stati installati utilizzando la AWS ParallelCluster versione 3.0.0 e successive con il job scheduler. Slurm Per ulteriori informazioni sulla configurazione di più code, vedere. [Configurazione di più code](#)

Se uno dei cluster in esecuzione presenta problemi, posiziona il cluster in uno STOPPED stato eseguendo il comando seguente prima di iniziare la risoluzione dei problemi. In questo modo si evitano costi imprevisti.

```
$ pcluster update-compute-fleet --cluster-name mycluster \  
  --status STOP_REQUESTED
```

È possibile elencare i flussi di log disponibili dai nodi del cluster utilizzando il [pcluster list-cluster-log-streams](#) comando e il filtraggio utilizzando uno `private-dns-name` dei nodi in errore o il nodo principale:

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \  
  --filters 'Name=private-dns-name,Values=ip-10-0-0-101'
```

È quindi possibile recuperare il contenuto del flusso di log per analizzarlo utilizzando il [pcluster get-cluster-log-events](#) comando e passando il `--log-stream-name` corrispondente a uno dei log chiave indicati nella sezione seguente:

```
$ pcluster get-cluster-log-events --cluster-name mycluster \  
  --region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init
```

AWS ParallelCluster crea flussi di CloudWatch log del cluster in gruppi di log. È possibile visualizzare questi registri nei dashboard personalizzati o nei CloudWatch gruppi di log della console. Per ulteriori informazioni, consultare [Integrazione con Amazon CloudWatch Logs](#) e [CloudWatch Pannello di controllo Amazon](#).

Argomenti

- [Registri chiave per il debug](#)
- [Viene visualizzato un `InsufficientInstanceCapacity` errore `slurm\_resume.log` quando non riesco a eseguire un processo o `clustermgtd.log` quando non riesco a creare un cluster](#)
- [Risoluzione dei problemi di inizializzazione dei nodi](#)
- [Risoluzione dei problemi di sostituzioni e terminazioni impreviste dei nodi](#)
- [Sostituzione, interruzione o spegnimento di istanze e nodi problematici](#)
- [Stato della coda \(partizione\) `Inactive`](#)
- [Risoluzione di altri problemi noti relativi a nodi e processi](#)

Registri chiave per il debug

La tabella seguente fornisce una panoramica dei log chiave per il nodo principale:

- `/var/log/cfn-init.log`- Questo è il registro di CloudFormation inizializzazione. Contiene tutti i comandi che sono stati eseguiti durante la configurazione di un'istanza. Usalo per risolvere i problemi di inizializzazione.
- `/var/log/chef-client.log`- Questo è il registro del client Chef. Contiene tutti i comandi che sono stati eseguiti tramite Chef/CINC. Usalo per risolvere i problemi di inizializzazione.
- `/var/log/parallelcluster/slurm_resume.log`- Questo è un registro. ResumeProgram Lancia istanze per nodi dinamici. Usalo per risolvere i problemi di avvio dei nodi dinamici.
- `/var/log/parallelcluster/slurm_suspend.log`- Questo è il registro. SuspendProgram Viene chiamato quando le istanze vengono terminate per i nodi dinamici. Usalo per risolvere i problemi di terminazione dei nodi dinamici. Quando controlli questo registro, dovresti controllare anche il registro. `clustermgtd`
- `/var/log/parallelcluster/clustermgtd`- Questo è il `clustermgtd` registro. Funziona come il demone centralizzato che gestisce la maggior parte delle azioni operative del cluster. Usalo per risolvere eventuali problemi di avvio, interruzione o funzionamento del cluster.
- `/var/log/slurmctld.log`- Questo è il registro del demone di Slurm controllo. AWS ParallelCluster non prende decisioni di scalabilità. Piuttosto, tenta solo di avviare risorse per soddisfare i Slurm requisiti. È utile per problemi di scalabilità e allocazione, problemi relativi al lavoro e qualsiasi problema relativo all'avvio e alla cessazione della pianificazione.
- `/var/log/parallelcluster/compute_console_output`- Questo registro registra l'output della console da un sottoinsieme campione di nodi di calcolo statici che sono terminati in modo imprevisto. Utilizza questo registro se i nodi di calcolo statici terminano e i log dei nodi di calcolo non sono disponibili in. CloudWatch Il `compute_console_output` log contenuto che ricevi è lo stesso quando usi la console Amazon EC2 o AWS CLI per recuperare l'output della console dell'istanza.

Questi sono i log chiave per i nodi di calcolo:

- `/var/log/cloud-init-output.log`- Questo è il log [cloud-init](#). Contiene tutti i comandi che sono stati eseguiti durante la configurazione di un'istanza. Usalo per risolvere i problemi di inizializzazione.
- `/var/log/parallelcluster/computemgtd`- Questo è il registro. `computemgtd` Viene eseguito su ogni nodo di elaborazione per monitorare il nodo nel raro caso in cui il `clustermgtd` demone sul nodo principale sia offline. Usalo per risolvere problemi di terminazione imprevisti.
- `/var/log/slurmd.log`- Questo è il registro del demone di Slurm calcolo. Usalo per risolvere problemi di inizializzazione e errori di calcolo.

Viene visualizzato un **InsufficientInstanceCapacity** errore **slurm_resume.log** quando non riesco a eseguire un processo o **clustermgtd.log** quando non riesco a creare un cluster

Se il cluster utilizza uno Slurm scheduler, si verifica un problema di capacità insufficiente. Se non ci sono abbastanza istanze disponibili quando viene effettuata una richiesta di avvio dell'istanza, viene restituito un **InsufficientInstanceCapacity** errore.

Per quanto riguarda la capacità statica delle istanze, è possibile trovare l'errore nel **clustermgtd** registro all'indirizzo `/var/log/parallelcluster/clustermgtd`.

Per quanto riguarda la capacità dinamica delle istanze, è possibile trovare l'errore nel **ResumeProgram** registro all'indirizzo `/var/log/parallelcluster/slurm_resume.log`.

Il messaggio è simile al seguente esempio:

```
An error occurred (InsufficientInstanceCapacity) when calling the RunInstances/
CreateFleet operation...
```

In base al tuo caso d'uso, valuta la possibilità di utilizzare uno dei seguenti metodi per evitare di ricevere questi tipi di messaggi di errore:

- Disattivate il gruppo di posizionamento, se abilitato. Per ulteriori informazioni, consulta [Gruppi di posizionamento e problemi di avvio delle istanze](#).
- Riserva la capacità per le istanze e avviale con ODCR (On-Demand Capacity Reservations). Per ulteriori informazioni, consulta [Avvio di istanze con On-Demand Capacity Reservations \(ODCR\)](#).
- Configura più risorse di elaborazione con diversi tipi di istanze. Se il tuo carico di lavoro non richiede un tipo di istanza specifico, puoi sfruttare un failover rapido e insufficiente con più risorse di elaborazione. Per ulteriori informazioni, consulta [Slurmfailover rapido con capacità insufficiente del cluster](#).
- Configura più tipi di istanze nella stessa risorsa di elaborazione e sfrutta l'allocazione di più tipi di istanza. Per ulteriori informazioni sulla configurazione di più istanze, consulta [Allocazione di più tipi di istanza con Slurm](#) e///. [SchedulingSlurmQueuesComputeResourcesInstances](#)
- Sposta la coda in una zona di disponibilità diversa modificando l'ID della sottorete nella configurazione [Scheduling](#) del cluster///. [SlurmQueuesNetworkingSubnetIds](#)

- Se il tuo carico di lavoro non è strettamente collegato, distribuisce la coda tra diverse zone di disponibilità. Per ulteriori informazioni sulla configurazione di più sottoreti, consulta [SchedulingSlurmQueuesNetworkingSubnetIds](#).

Risoluzione dei problemi di inizializzazione dei nodi

Questa sezione illustra come risolvere i problemi di inizializzazione dei nodi. Ciò include i problemi in cui il nodo non riesce ad avviarsi, accendersi o entrare a far parte di un cluster.

Argomenti

- [Nodo principale](#)
- [Nodi di calcolo](#)

Nodo principale

Registri applicabili:

- `/var/log/cfn-init.log`
- `/var/log/chef-client.log`
- `/var/log/parallelcluster/clustermgtd`
- `/var/log/parallelcluster/slurm_resume.log`
- `/var/log/slurmctld.log`

Controlla i `/var/log/chef-client.log` `log /var/log/cfn-init.log` and o i flussi di log corrispondenti. Questi registri contengono tutte le azioni eseguite durante la configurazione del nodo principale. La maggior parte degli errori che si verificano durante l'installazione dovrebbe contenere messaggi di errore nel `/var/log/chef-client.log` registro. Se `OnNodeStart` nella configurazione del cluster sono specificati i nostri `OnNodeConfigured` script, ricontrolla che lo script venga eseguito correttamente tramite i messaggi di registro.

Quando viene creato un cluster, il nodo principale deve attendere che i nodi di calcolo si uniscano al cluster prima di potersi unire al cluster. Per questo motivo, se i nodi di elaborazione non riescono a unirsi al cluster, anche il nodo principale fallisce. È possibile seguire una di queste serie di procedure, a seconda del tipo di note di calcolo utilizzate, per risolvere questo tipo di problema:

Nodi di calcolo

- Registri applicabili:
 - `/var/log/cloud-init-output.log`
 - `/var/log/slurmd.log`
- Se viene avviato un nodo di calcolo/`/var/log/cloud-init-output.log`, controllate innanzitutto che contenga i log di configurazione simili a quelli del `/var/log/chef-client.log` nodo principale. La maggior parte degli errori che si verificano durante l'installazione dovrebbero contenere messaggi di errore nel registro. `/var/log/cloud-init-output.log` Se nella configurazione del cluster sono specificati script di preinstallazione o post-installazione, verificate che siano stati eseguiti correttamente.
- Se utilizzi un'AMI personalizzata con modifiche alla Slurm configurazione, potrebbe esserci un errore Slurm correlato che impedisce al nodo di calcolo di entrare a far parte del cluster. Per gli errori relativi allo scheduler, controlla il registro. `/var/log/slurmd.log`

Nodi di calcolo dinamici:

- Cerca in ResumeProgram log (`/var/log/parallelcluster/slurm_resume.log`) il nome del tuo nodo di calcolo per vedere se ResumeProgram è mai stato chiamato con il nodo. (Se ResumeProgram non è mai stato chiamato, puoi controllare `slurmctld` log (`/var/log/slurmctld.log`) per determinare se hai Slurm mai provato a chiamare ResumeProgram con il nodo).
- Tieni presente che autorizzazioni errate per ResumeProgram potrebbero causare ResumeProgram un errore silenzioso. Se utilizzi un'AMI personalizzata con modifiche alla ResumeProgram configurazione, verifica che sia di proprietà dell'`slurmutente` e disponga dell'autorizzazione `744 (rwxr--r--)`. ResumeProgram
- Se ResumeProgram viene chiamato, controlla se è stata avviata un'istanza per il nodo. Se non è stata avviata alcuna istanza, viene visualizzato un messaggio di errore che descrive l'errore di avvio.
- Se l'istanza viene avviata, potrebbe essersi verificato un problema durante il processo di configurazione. Dovresti vedere l'indirizzo IP privato e l'ID dell'istanza corrispondenti dal ResumeProgram registro. Inoltre, puoi consultare i registri di configurazione corrispondenti per l'istanza specifica. Per ulteriori informazioni sulla risoluzione di un errore di configurazione con un nodo di calcolo, consulta la sezione successiva.

Nodi di calcolo statici:

- Controlla il registro `clustermgtd (/var/log/parallelcluster/clustermgtd)` per vedere se sono state lanciate istanze per il nodo. Se non sono state avviate, dovrebbe apparire un messaggio di errore chiaro che descrive in dettaglio l'errore di avvio.
- Se l'istanza viene avviata, c'è qualche problema durante il processo di configurazione. Dovresti vedere l'indirizzo IP privato e l'ID dell'istanza corrispondenti dal `ResumeProgram` registro. Inoltre, puoi consultare i registri di configurazione corrispondenti per l'istanza specifica.

Nodi di calcolo supportati da istanze Spot:

- Se è la prima volta che utilizzi le istanze Spot e il lavoro rimane in un PD (stato in sospeso), ricontrolla il file. `/var/log/parallelcluster/slurm_resume.log` Probabilmente troverai un errore come il seguente:

```
2022-05-20 13:06:24,796 - [slurm_plugin.common:add_instances_for_nodes] - ERROR -  
Encountered exception when launching instances for nodes (x1) ['spot-dy-t2micro-2']:  
An error occurred (AuthFailure.ServiceLinkedRoleCreationNotPermitted) when calling  
the RunInstances operation: The provided credentials do not have permission to  
create the service-linked role for Amazon EC2 Spot Instances.
```

Quando utilizzi le istanze Spot, nel tuo account deve esistere un ruolo

`AWSServiceRoleForEC2Spot` collegato al servizio. Per creare questo ruolo nel tuo account utilizzando AWS CLI, esegui il seguente comando:

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Per ulteriori informazioni, consulta [Utilizzo di Istanze spot](#) la Guida per l' AWS ParallelCluster utente e il [ruolo collegato ai servizi per le richieste di istanze Spot](#) nella Guida per l'utente di Amazon EC2.

Risoluzione dei problemi di sostituzioni e terminazioni impreviste dei nodi

Questa sezione continua a esplorare come risolvere i problemi relativi ai nodi, in particolare quando un nodo viene sostituito o terminato in modo imprevisto.

- Registri applicabili:

- `/var/log/parallelcluster/clustermgtd`(nodo principale)
- `/var/log/slurmctld.log`(nodo principale)
- `/var/log/parallelcluster/computemgtd`(nodo di calcolo)

Nodi sostituiti o terminati in modo imprevisto

- Controlla nel `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) per vedere se un nodo `clustermgtd` è stato sostituito o terminato. Nota che `clustermgtd` gestisce tutte le normali azioni di manutenzione del nodo.
- Se il nodo `clustermgtd` viene sostituito o terminato, dovrebbe esserci un messaggio che spiega in dettaglio il motivo per cui è stata intrapresa questa azione sul nodo. Se il motivo è correlato allo scheduler (ad esempio, perché il nodo è attivoDOWN), controlla il `slurmctld` log in per ulteriori informazioni. Se il motivo è correlato ad Amazon EC2, dovrebbe esserci un messaggio informativo che descriva in dettaglio il problema relativo ad Amazon EC2 che ha richiesto la sostituzione.
- Se `clustermgtd` non hai terminato il nodo, verifica innanzitutto se si trattava di una terminazione prevista da parte di Amazon EC2, in particolare di una terminazione spot. `computemgtd`, in esecuzione su un nodo di elaborazione, può anche terminare un nodo se viene considerato non integro. Controlla `computemgtd` log (`/var/log/parallelcluster/computemgtd`) per vedere se il nodo è `computemgtd` terminato.

Nodi falliti

- Controlla `slurmctld` log (`/var/log/slurmctld.log`) per vedere perché un job o un nodo non sono riusciti. Tieni presente che i lavori vengono automaticamente messi in coda in caso di errore di un nodo.
- Se `slurm_resume` segnala che il nodo è stato avviato e dopo alcuni minuti `clustermgtd` segnala che non esiste un'istanza corrispondente in Amazon EC2 per quel nodo, il nodo potrebbe fallire durante la configurazione. Per recuperare il log da un compute (`/var/log/cloud-init-output.log`), procedi nel seguente modo:
 - Invia un lavoro per consentire la creazione Slurm di un nuovo nodo.
 - Attendi l'avvio del nodo di calcolo.
 - Modifica il comportamento di chiusura avviato dall'istanza in modo che un nodo di calcolo guasto venga interrotto anziché terminato.

```
$ aws ec2 modify-instance-attribute \
```



```
--instance-id i-1234567890abcdef0 \  
--instance-initiated-shutdown-behavior "{\"Value\": \"stop\"}"
```

- Abilita la protezione da cessazione.

```
$ aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --disable-api-termination
```

- Contrassegna il nodo in modo che sia facilmente identificabile.

```
$ aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=Name,Value=QUARANTINED-Compute
```

- Stacca il nodo dal cluster modificando il `parallelcluster:cluster-name` tag.

```
$ aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=parallelcluster:clustername,Value=QUARANTINED-ClusterName
```

- Recupera l'output della console dal nodo con questo comando.

```
$ aws ec2 get-console-output --instance-id i-1234567890abcdef0 --output text
```

Sostituzione, interruzione o spegnimento di istanze e nodi problematici

- Registri applicabili:
 - `/var/log/parallelcluster/clustermgtd`(nodo principale)
 - `/var/log/parallelcluster/slurm_suspend.log`(nodo principale)
- Nella maggior parte dei casi, `clustermgtd` gestisce tutte le azioni di terminazione previste dell'istanza. Controlla il `clustermgtd` registro per vedere perché non è riuscito a sostituire o terminare un nodo.
- Se i nodi dinamici non funzionano [SlurmSettingsProprietà](#) correttamente, controlla il `SuspendProgram` registro per vedere se `SuspendProgram` è stato chiamato usando `slurmctl` il nodo specifico come argomento. Nota che in realtà `SuspendProgram` non esegue alcuna azione. Piuttosto, registra solo quando viene chiamato. La terminazione e il `NodeAdd`

ripristino di tutte le istanze vengono eseguiti da `clustermgtd` Slurm riporta `SuspendTimeout` automaticamente i nodi in uno `POWER_SAVING` stato.

- Se i nodi di calcolo falliscono continuamente a causa di errori di bootstrap, verifica se vengono avviati con `enabled`. [Slurm modalità protetta dal cluster](#) Se la modalità protetta non è abilitata, modifica le impostazioni della modalità protetta per abilitare la modalità protetta. Risolvi i problemi e correggi lo script di bootstrap.

Stato della coda (partizione) **Inactive**

Se si esegue `sinfo` e l'output mostra code con `AVAIL` stato di `inact`, è possibile che il cluster sia [Slurm modalità protetta dal cluster](#) abilitato e che la coda sia stata impostata sullo `INACTIVE` stato per un periodo di tempo predefinito.

Risoluzione di altri problemi noti relativi a nodi e processi

Un altro tipo di problema noto è che AWS ParallelCluster potrebbe non riuscire ad allocare i lavori o a prendere decisioni sulla scalabilità. Con questo tipo di problema, avvia, termina o gestisce le risorse AWS ParallelCluster solo in base alle istruzioni. Slurm Per questi problemi, controlla il `slurmctld` registro per risolverli.

Gruppi di posizionamento e problemi di avvio delle istanze

Per ottenere la latenza internodale più bassa, utilizzate un gruppo di posizionamento. Un gruppo di collocamento garantisce che le istanze si trovino sulla stessa spina dorsale di rete. Se non ci sono abbastanza istanze disponibili quando viene effettuata una richiesta, viene restituito un `InsufficientInstanceCapacity` errore. Per ridurre la possibilità di ricevere questo errore quando si utilizzano i gruppi di posizionamento dei cluster, impostate il [Enabled](#) parametro [SlurmQueuesNetworking/PlacementGroup](#)//`sufalse`.

Per un maggiore controllo sull'accesso alla capacità, prendi in considerazione l'[avvio di istanze con ODCR \(On-Demand Capacity Reservations\)](#).

Per ulteriori informazioni, consulta [Risoluzione dei problemi di avvio delle istanze](#) e [Ruoli e limitazioni dei gruppi di posizionamento](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Sostituzione delle directory

Alcune directory non possono essere sostituite. Se hai problemi a sostituire la directory, potrebbe essere così. Le seguenti directory sono condivise tra i nodi e non possono essere sostituite.

- `/opt/intel-` Sono inclusi Intel MPI, Intel Parallel Studio e file correlati.
- `/opt/slurm-` Sono inclusi Slurm Workload Manager e file correlati. (Condizionale, solo se `Scheduler: slurm.`)

Risoluzione dei problemi in Amazon DCV

Argomenti

- [Registri per Amazon DCV](#)
- [Problemi con Ubuntu Amazon DCV](#)

Registri per Amazon DCV

I log di Amazon DCV vengono scritti nei file della `/var/log/dcv/` directory. La revisione di questi registri può aiutare a risolvere i problemi.

Il tipo di istanza deve avere almeno 1,7 gibibyte (GiB) di RAM per eseguire Amazon DCV. I tipi di istanze nano e micro non dispongono di memoria sufficiente per eseguire Amazon DCV.

AWS ParallelCluster crea flussi di log Amazon DCV in gruppi di log. Puoi visualizzare questi log nei dashboard personalizzati o nei gruppi di log della CloudWatch console. Per ulteriori informazioni, consultare [Integrazione con Amazon CloudWatch Logs](#) e [CloudWatch Pannello di controllo Amazon](#).

Problemi con Ubuntu Amazon DCV

Quando esegui Gnome Terminal su una sessione Amazon DCV su Ubuntu, potresti non avere automaticamente accesso all'ambiente utente AWS ParallelCluster reso disponibile tramite la shell di accesso. L'ambiente utente fornisce moduli di ambiente come `openmpi` o `intelmpi` e altre impostazioni utente.

Le impostazioni predefinite di Gnome Terminal impediscono alla shell di avviarsi come shell di accesso. Ciò significa che i profili della shell non vengono generati automaticamente e l'ambiente AWS ParallelCluster utente non viene caricato.

Per creare correttamente il profilo shell e accedere all'ambiente AWS ParallelCluster utente, effettuate una delle seguenti operazioni:

- Modificate le impostazioni predefinite del terminale:
 1. Scegli il menu Modifica nel terminale Gnome.
 2. Seleziona Preferenze, quindi Profili.
 3. Scegli Comando e seleziona Esegui comando come shell di accesso.
 4. Apri un nuovo terminale.
- Usa la riga di comando per trovare i profili disponibili:

```
$ source /etc/profile && source $HOME/.bashrc
```

Risoluzione dei problemi nei cluster con integrazione AWS Batch

Questa sezione fornisce possibili suggerimenti per la risoluzione dei problemi per i cluster con integrazione dello AWS Batch scheduler, in particolare per quanto riguarda i problemi relativi ai nodi principali, ai problemi di elaborazione, agli errori dei processi e agli errori di timeout.

Argomenti

- [Problemi relativi al nodo principale](#)
- [Problemi di calcolo](#)
- [Errori del processo](#)
- [Errore Connect timeout sull'URL dell'endpoint](#)

Problemi relativi al nodo principale

È possibile risolvere i problemi di configurazione del nodo principale allo stesso modo di un Slurm cluster (ad eccezione dei log Slurm specifici). Per ulteriori informazioni su questi problemi, consulta [Nodo principale](#).

Problemi di calcolo

AWS Batch gestisce gli aspetti di scalabilità e calcolo dei tuoi servizi. Se riscontri problemi relativi all'elaborazione, consulta la documentazione AWS Batch [sulla risoluzione dei problemi per ricevere assistenza](#).

Errori del processo

Se un processo fallisce, puoi eseguire il [awsbout](#) comando per recuperare l'output del processo. Puoi anche eseguire il [awsbstat](#) comando per ottenere un collegamento ai log dei lavori archiviati da Amazon CloudWatch.

Errore Connect timeout sull'URL dell'endpoint

Se i lavori paralleli multinodo falliscono e restituiscono un errore: `Connect timeout on endpoint URL`

- Nel log `awsbout` di output, verificate che il job sia parallelo a più nodi rispetto all'output: `Detected 3/3 compute nodes. Waiting for all compute nodes to start.`
- Verifica se la sottorete dei nodi di calcolo è pubblica.

I lavori paralleli multinodo non supportano l'uso di sottoreti pubbliche quando si utilizza in. AWS Batch AWS ParallelCluster Usa una sottorete privata per i nodi e i lavori di elaborazione. Per ulteriori informazioni, consulta [Considerazioni sull'ambiente di calcolo](#) nella Guida per l'utente. AWS Batch Per configurare una sottorete privata per i nodi di calcolo, consulta. [AWS ParallelCluster con scheduler AWS Batch](#)

Risoluzione dei problemi di integrazione multiutente con Active Directory

Questa sezione è rilevante per i cluster integrati con Active Directory.

Se la funzionalità di integrazione di Active Directory non funziona come previsto, i log SSSD possono fornire informazioni diagnostiche utili. Questi registri si trovano nei nodi del cluster `/var/log/sss`. Per impostazione predefinita, vengono archiviati anche nel gruppo di CloudWatch log Amazon di un cluster.

Argomenti

- [Risoluzione dei problemi specifici di Active Directory](#)
- [Abilita la modalità di debug](#)
- [Come passare da LDAPS a LDAP](#)
- [Come disabilitare la verifica dei certificati del server LDAPS](#)

- [Come accedere con una chiave SSH anziché una password](#)
- [Come reimpostare una password utente e le password scadute](#)
- [Come verificare il dominio aggiunto](#)
- [Come risolvere i problemi relativi ai certificati](#)
- [Come verificare che l'integrazione con Active Directory funzioni](#)
- [Come risolvere i problemi di accesso ai nodi di calcolo](#)
- [Problemi noti con i job SimCenter StarCCM+ in un ambiente multiutente](#)
- [Problemi noti relativi alla risoluzione dei nomi utente](#)
- [Come risolvere i problemi di creazione della home directory](#)

Risoluzione dei problemi specifici di Active Directory

Questa sezione è rilevante per la risoluzione dei problemi specifici di un tipo di Active Directory.

Simple AD

- Il `DomainReadOnlyUser` valore deve corrispondere alla ricerca di base della directory Simple AD per gli utenti:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

Nota `cn` per `Users`.

- L'utente amministratore predefinito è `Administrator`.
- `Ldapsearch` richiede il nome NetBIOS prima del nome utente.

`Ldapsearch` la sintassi deve essere la seguente:

```
$ ldapsearch -x -D "corp\\Administrator" -w "Password" -H ldap://192.0.2.103 \
-b "cn=Users,dc=corp,dc=example,dc=com"
```

AWS Managed Microsoft AD

- Il `DomainReadOnlyUser` valore deve corrispondere alla ricerca degli utenti AWS Managed Microsoft AD nella base della directory:

```
cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

- L'utente amministratore predefinito è Admin.
- Ldapsearch la sintassi deve essere la seguente:

```
$ ldapsearch -x -D "Admin" -w "Password" -H ldap://192.0.2.103 \
  -b "ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

Abilita la modalità di debug

I log di debug da SSSD possono essere utili per risolvere i problemi. Per abilitare la modalità di debug, è necessario aggiornare il cluster con le seguenti modifiche apportate alla configurazione del cluster:

```
DirectoryService:
  AdditionalSssdConfigs:
    debug_level: "0x1ff"
```

Come passare da LDAPS a LDAP

Il passaggio da LDAPS (LDAP con TLS/SSL) a LDAP è sconsigliato perché LDAP da solo non fornisce alcuna crittografia. Tuttavia, può essere utile per scopi di test e risoluzione dei problemi.

È possibile ripristinare la configurazione precedente del cluster aggiornando il cluster con la definizione di configurazione precedente.

Per passare da LDAPS a LDAP, è necessario aggiornare il cluster con le seguenti modifiche nella configurazione del cluster:

```
DirectoryService:
  LdapTlsReqCert: never
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

Come disabilitare la verifica dei certificati del server LDAPS

Può essere utile disabilitare temporaneamente la verifica del certificato del server LDAPS sul nodo principale, a scopo di test o risoluzione dei problemi.

È possibile ripristinare il cluster alla configurazione precedente aggiornando il cluster con la definizione di configurazione precedente.

Per disabilitare la verifica del certificato del server LDAPS, è necessario aggiornare il cluster con le seguenti modifiche nella configurazione del cluster:

```
DirectoryService:
  LdapTlsReqCert: never
```

Come accedere con una chiave SSH anziché una password

La chiave SSH viene creata `/home/$user/.ssh/id_rsa` dopo il primo accesso con una password. Per accedere con la chiave SSH, è necessario accedere con la password, copiare la chiave SSH localmente e quindi utilizzarla senza password SSH come al solito:

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY $username@$head_node_ip
```

Come reimpostare una password utente e le password scadute

Se un utente perde l'accesso a un cluster, [AWS Managed Microsoft AD la sua password potrebbe essere scaduta](#).

Per reimpostare la password, esegui il comando seguente con un utente e un ruolo con autorizzazione di scrittura sulla directory:

```
$ aws ds reset-user-password \
  --directory-id "d-abcdef01234567890" \
  --user-name "USER_NAME" \
  --new-password "NEW_PASSWORD" \
  --region "region-id"
```

Se si reimposta la password per [DirectoryService/DomainReadOnlyUser](#):

1. Assicurati di aggiornare [DirectoryService/PasswordSecretArn](#)secret con la nuova password.
2. Aggiorna il cluster per il nuovo valore segreto:
 - a. Arresta la flotta di calcolo con il `pcluster update-compute-fleet` comando.
 - b. Esegui il comando seguente dall'interno del nodo principale del cluster.

```
$ sudo /opt/parallelcluster/scripts/directory_service/
update_directory_service_password.sh
```


Dopo la reimpostazione della password e l'aggiornamento del cluster, è necessario ripristinare l'accesso al cluster dell'utente.

Per ulteriori informazioni, vedere [Reimpostazione di una password utente](#) nella Guida all'Directory Service amministrazione.

Come verificare il dominio aggiunto

Il comando seguente deve essere eseguito da un'istanza aggiunta al dominio, non dal nodo principale.

```
$ realm list corp.example.com \  
type: kerberos \  
realm-name: CORP.EXAMPLE.COM \  
domain-name: corp.example.com \  
configured: kerberos-member \  
server-software: active-directory \  
client-software: sssd \  
required-package: oddjob \  
required-package: oddjob-mkhomedir \  
required-package: sssd \  
required-package: adcli \  
required-package: samba-common-tools \  
login-formats: %U \  
login-policy: allow-realm-logins
```

Come risolvere i problemi relativi ai certificati

Quando la comunicazione LDAPS non funziona, può essere dovuto a errori nella comunicazione TLS, che a loro volta possono essere dovuti a problemi con i certificati.

Note sui certificati:

- Il certificato specificato nella configurazione del cluster `LdapTlsCaCert` deve essere un pacchetto di certificati PEM contenente i certificati per l'intera catena di certificati di autorità (CA) che ha emesso i certificati per i controller di dominio.
- Un pacchetto di certificati PEM è un file composto dalla concatenazione di certificati PEM.
- Un certificato in formato PEM (generalmente utilizzato in Linux) è equivalente a un certificato in formato DER base64 (in genere esportato da Windows).

- Se il certificato per i controller di dominio è emesso da una CA subordinata, il pacchetto di certificati deve contenere il certificato sia della CA subordinata che della CA principale.

Fasi di verifica della risoluzione dei problemi:

I seguenti passaggi di verifica presuppongono che i comandi vengano eseguiti dall'interno del nodo principale del cluster e che il controller di dominio sia raggiungibile all'indirizzo **SERVER:PORT**.

Per risolvere un problema relativo ai certificati, segui questi passaggi di verifica:

Passaggi di verifica:

1. Verifica la connessione ai controller di dominio Active Directory:

Verifica di poterti connettere a un controller di dominio. Se questo passaggio ha esito positivo, la connessione SSL al controller di dominio ha esito positivo e il certificato viene verificato. Il tuo problema non è correlato ai certificati.

Se questo passaggio fallisce, procedi con la verifica successiva.

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

2. Controlla la verifica del certificato:

Verifica che il pacchetto di certificati CA locale sia in grado di convalidare il certificato fornito dal controller di dominio. Se questo passaggio ha esito positivo, il problema non è correlato ai certificati, ma ad altri problemi di rete.

Se questo passaggio fallisce, procedi con la verifica successiva.

```
$ openssl verify -verbose -  
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

3. Controlla il certificato fornito dai controller di dominio Active Directory:

Verifica che il contenuto del certificato fornito dai controller di dominio sia quello previsto. Se questo passaggio ha esito positivo, probabilmente hai problemi con il certificato CA utilizzato per verificare i controller. Vai al passaggio successivo per la risoluzione dei problemi.

Se questo passaggio non riesce, è necessario correggere il certificato emesso per i controller di dominio e rieseguire la procedura di risoluzione dei problemi.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

4. Controlla il contenuto di un certificato:

Verifica che il contenuto del certificato fornito dai controller di dominio sia quello previsto. Se questo passaggio ha esito positivo, probabilmente hai problemi con il certificato CA utilizzato per verificare il controller, vai al passaggio successivo per la risoluzione dei problemi.

Se questo passaggio non riesce, è necessario correggere il certificato emesso per i controller di dominio ed eseguire nuovamente la procedura di risoluzione dei problemi.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

5. Controlla il contenuto del pacchetto di certificati CA locale:

Verifica che il contenuto del pacchetto di certificati CA locale utilizzato per convalidare il certificato dei controller di dominio sia quello previsto. Se questo passaggio ha esito positivo, probabilmente hai problemi con il certificato fornito dai controller di dominio.

Se questo passaggio non riesce, è necessario correggere il pacchetto di certificati CA emesso per i controller di dominio ed eseguire nuovamente la procedura di risoluzione dei problemi.

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

Come verificare che l'integrazione con Active Directory funzioni

Se i due controlli seguenti hanno esito positivo, l'integrazione con Active Directory funziona.

Controlli:

1. Puoi scoprire gli utenti definiti nella directory:

Dall'interno del nodo principale del cluster, come `ec2-user`:

```
$ getent passwd $ANY_AD_USER
```

2. È possibile accedere al nodo principale tramite SSH fornendo la password dell'utente:

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

Se il primo controllo fallisce, ci aspettiamo che anche il controllo due fallisca.

Controlli aggiuntivi per la risoluzione dei problemi:

- Verifica che l'utente esista nella directory.
- Abilita la [registrazione di debug](#).
- Valuta la possibilità di disabilitare temporaneamente la crittografia [passando da LDAPS a LDAP per escludere problemi LDAPS](#).

Come risolvere i problemi di accesso ai nodi di calcolo

Questa sezione è rilevante per l'accesso ai nodi di calcolo nei cluster integrati con Active Directory.

Con AWS ParallelCluster, gli accessi tramite password ai nodi di calcolo del cluster sono disabilitati in base alla progettazione.

Tutti gli utenti devono utilizzare la propria chiave SSH per accedere ai nodi di calcolo.

Gli utenti possono recuperare la propria chiave SSH nel nodo principale dopo la prima autenticazione (ad esempio il login), se [GenerateSshKeysForUsers](#) è abilitata nella configurazione del cluster.

Quando gli utenti si autenticano sul nodo principale per la prima volta, possono recuperare le chiavi SSH che vengono generate automaticamente per loro come utenti della directory. Vengono inoltre create le home directory per l'utente. Ciò può accadere anche la prima volta che un sudo-user passa a un utente nel nodo principale.

Se un utente non ha effettuato l'accesso al nodo principale, le chiavi SSH non vengono generate e l'utente non sarà in grado di accedere ai nodi di calcolo.

Problemi noti con i job SimCenter StarCCM+ in un ambiente multiutente

Questa sezione è relativa ai lavori avviati in un ambiente multiutente dal software di fluidodinamica computazionale Simcenter StarCCM+ di Siemens.

Se si eseguono processi StarCCM+ v16 configurati per utilizzare l'IntelMPI integrato, per impostazione predefinita i processi MPI vengono avviati tramite SSH.

A causa di un [Slurmbug](#) noto che causa una risoluzione errata del nome utente, i processi potrebbero fallire con un errore del tipo. `error setting up the bootstrap proxies` Questo bug riguarda solo le AWS ParallelCluster versioni 3.1.1 e 3.1.2.

Per evitare che ciò accada, forza IntelMPI a utilizzare IntelMPI come metodo di bootstrap MPI. Slurm [Esporta la variabile di ambiente I_MPI_HYDRA_BOOTSTRAP=slurm nello script di lavoro che avvia StarCCM+, come descritto nella documentazione ufficiale di IntelMPI.](#)

Problemi noti relativi alla risoluzione dei nomi utente

Questa sezione è rilevante per il recupero dei nomi utente all'interno dei job.

A causa di un [bug noto in Slurm](#), il nome utente recuperato all'interno di un processo di lavoro potrebbe essere nobody se si esegue un lavoro senza. `srn` Questo bug riguarda solo AWS ParallelCluster le versioni 3.1.1 e 3.1.2.

Ad esempio, se si esegue il comando `sbatch --wrap 'srn id'` come utente della directory, viene restituito il nome utente corretto. Tuttavia, se lo si esegue `sbatch --wrap 'id'` come utente della directory, nobody potrebbe essere restituito come nome utente.

È possibile utilizzare le seguenti soluzioni alternative.

1. Avvia il tuo lavoro con `'srn'` invece di `'sbatch'`, se possibile.
2. Abilita l'enumerazione SSSD impostando la configurazione [AdditionalSssdConfigs](#) nel cluster come segue.

```
AdditionalSssdConfigs:
  enumerate: true
```

Come risolvere i problemi di creazione della home directory

Questa sezione è rilevante per i problemi di creazione della home directory.

Se vedi errori come quello mostrato nell'esempio seguente, significa che non è stata creata una home directory quando hai effettuato il primo accesso al nodo principale. Oppure, una home directory non è stata creata automaticamente quando sei passato per la prima volta da un sudoer a un utente di Active Directory nel nodo principale.

```
$ ssh AD_USER@$HEAD_NODE_IP
```

```
/opt/parallelcluster/scripts/generate_ssh_key.sh failed: exit code 1
```

```

  _|  _|_ )
  _| (    /  Amazon Linux 2 AMI
 _|\___|___|

```

```
https://aws.amazon.com/amazon-linux-2/
```

```
Could not chdir to home directory /home/PclusterUser85: No such file or directory
```

L'errore di creazione della home directory può essere causato dai `oddjob-mkhomedir` pacchetti `oddjob` and installati nel nodo principale del cluster.

Senza una home directory e una chiave SSH, l'utente non può inviare job o SSH nei nodi del cluster.

Se hai bisogno dei `oddjob` pacchetti nel tuo sistema, verifica che il `oddjobd` servizio sia in esecuzione e aggiorna i file di configurazione PAM per assicurarti che la home directory sia stata creata. A tale scopo, eseguite i comandi nel nodo head, come illustrato nell'esempio seguente.

```
sudo systemctl start oddjobd
sudo authconfig --enablemkhomedir --updateall
```

Se non avete bisogno dei `oddjob` pacchetti nel vostro sistema, disinstallateli e aggiornate i file di configurazione PAM per assicurarvi che la home directory venga creata. Per fare ciò, esegui i comandi nel nodo head come mostrato nell'esempio seguente.

```
sudo yum remove -y oddjob oddjob-mkhomedir
sudo authconfig --enablemkhomedir --updateall
```

Risoluzione dei problemi relativi alle AMI personalizzate

Questa sezione fornisce possibili suggerimenti per la risoluzione di problemi relativi alle AMI personalizzate.

Quando usi un'AMI personalizzata, puoi visualizzare i seguenti avvisi:

```
"validationMessages": [
  {
    "level": "WARNING",
    "type": "CustomAmiTagValidator",
```

```

    "message": "The custom AMI may not have been created by pcluster. You can ignore
this warning if the AMI is shared or copied from another pcluster AMI. If the
AMI is indeed not created by pcluster, cluster creation will fail. If the cluster
creation fails, please go to https://docs.aws.amazon.com/parallelcluster/latest/ug/
troubleshooting.html#troubleshooting-stack-creation-failures for troubleshooting."
  },
  {
    "level": "WARNING",
    "type": "AmiOsCompatibleValidator",
    "message": "Could not check node AMI ami-0000012345 OS and cluster OS alinux2
compatibility, please make sure they are compatible before cluster creation and update
operations."
  }
]

```

Se sei sicuro che venga utilizzata l'AMI corretta, puoi ignorare questi avvisi.

Se non desideri visualizzare questi avvisi in futuro, contrassegna l'AMI personalizzata con i seguenti tag, dove si *my-os* trova una delle *alinux2*, *alinux2023ubuntu2404*, *ubuntu2204rhel8*, o *rhel9* e indica *"3.15.0"* la pcluster versione in uso:

```

$ aws ec2 create-tags \
  --resources ami-yourcustomAmi \
  --tags Key="parallelcluster:version",Value="3.15.0"
  Key="parallelcluster:os",Value="my-os"

```

Risoluzione dei problemi di un timeout di aggiornamento del cluster quando **cfn-hup** non è in esecuzione

L'helper cfn-hup è un daemon che rileva modifiche nei metadati della risorsa ed esegue azioni specificate dall'utente quando viene rilevata una modifica. In questo modo è possibile effettuare aggiornamenti di configurazione sulle istanze Amazon EC2 in esecuzione tramite l'UpdateStackazione API.

Attualmente il cfn-hup demone viene lanciato da. `supervisord` Ma dopo il lancio, il cfn-hup processo viene rimosso dal controllo. `supervisord` Se il cfn-hup demone viene ucciso da un attore esterno, non viene riavviato automaticamente. Se cfn-hup non è in esecuzione, durante un aggiornamento del cluster, lo CloudFormation stack avvia il processo di aggiornamento come previsto, ma la procedura di aggiornamento non viene attivata sul nodo principale e lo stack alla

fine entra in timeout. Dai log del cluster `/var/log/chef-client`, puoi vedere che la ricetta di aggiornamento non viene mai richiamata.

Controlla e riavvia **cfn-hup** in caso di guasti

1. Sul nodo principale, controlla se `cfn-hup` è in esecuzione:

```
$ ps aux | grep cfn-hup
```

2. Controlla `cfn-hup` il registro `/var/log/cfn-hup.log` e `/var/log/supervisord.log` il nodo principale.
3. Se `cfn-hup` non è in esecuzione, prova a riavviarlo eseguendo:

```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl  
start cfn-hup
```

Risoluzione dei problemi di rete

Questa sezione fornisce un suggerimento per la risoluzione dei problemi di rete, in particolare quando si tratta di un cluster in una singola sottorete pubblica.

Problemi relativi al cluster in un'unica sottorete pubblica

Controlla `cloud-init-output.log` da uno dei nodi di calcolo. Se trovi qualcosa come il seguente che indica che il nodo è bloccato nell'Slurminizializzazione, molto probabilmente è dovuto a un endpoint VPC DynamoDB mancante. Aggiungi l'endpoint DynamoDB. Per ulteriori informazioni, consulta [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#).

```
ruby_block[retrieve compute node info] action run[2022-03-11T17:47:11+00:00] INFO:  
Processing ruby_block[retrieve compute node info] action run (aws-parallelcluster-  
slurm::init line 31)
```

Aggiornamento del cluster non riuscito a causa di un'azione personalizzata **onNodeUpdated**

Quando uno [OnNodeUpdated](#) script [HeadNode/CustomActions](#) fallisce, l'aggiornamento fallisce e lo script non viene eseguito al momento del rollback. È responsabilità dell'utente eseguire

manualmente le operazioni di pulizia necessarie dopo il completamento del rollback. Ad esempio, se lo `OnNodeUpdated` script modifica lo stato di un campo in un file di configurazione (ad esempio, da `true` a `false`) e poi fallisce, è necessario ripristinare manualmente il valore del campo allo stato precedente all'aggiornamento (ad esempio, `false` to `true`). Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

Visualizzazione degli errori con la configurazione personalizzata Slurm

A partire dalla AWS ParallelCluster versione 3.6.0, non è più possibile scegliere come target singoli `prolog` o `epilog` script includendoli in una configurazione personalizzata. Slurm Nella AWS ParallelCluster versione 3.6.0 e nelle versioni successive, è necessario individuare gli script `prolog` e `epilog` gli script personalizzati nelle rispettive cartelle. Prolog Epilog Queste cartelle sono configurate per impostazione predefinita in modo che puntino a:

- Prolog punta a `/opt/slurm/etc/scripts/prolog.d/`.
- Epilog punta a `/opt/slurm/etc/scripts/epilog.d/`.

Ti consigliamo di mantenere invariati lo script di `90_plcluster_health_check_manager` prologo e lo script di `90_plcluster_noop` epilogo.

Slurm esegue gli script in ordine alfabetico inverso. Prolog Sia la Epilog cartella che deve contenere almeno un file. Per ulteriori informazioni, consultare [Slurm e prolog epilog](#) e [Slurm personalizzazione della configurazione](#).

Allarmi cluster

Il monitoraggio dello stato del cluster è essenziale per garantire prestazioni ottimali. AWS ParallelCluster consente di monitorare più allarmi CloudWatch basati sul nodo principale del cluster.

Questa sezione fornisce dettagli per ogni tipo di allarme del cluster Head Node, comprese le convenzioni di denominazione, le condizioni specifiche che attivano gli allarmi e i passaggi suggeriti per la risoluzione dei problemi.

La convenzione di denominazione per gli allarmi del cluster è, ad es. `CLUSTER_NAME-COMPONENT-METRIC` `mycluster-HeadNode-Cpu`

- `CLUSTER_NAME-HeadNode`: segnala lo stato generale del nodo principale. È rosso se almeno uno degli allarmi seguenti lo è.
- `CLUSTER_NAME-HeadNode-Health`: rosso se si verifica almeno un errore in Amazon EC2 Health Check. In caso di allarme, ti suggeriamo di dare un'occhiata a [Risoluzione dei problemi relativi alle istanze con](#) controlli di stato non riusciti.
- `CLUSTER_NAME-HeadNode-Cpu`: rosso se l'utilizzo della CPU è superiore al 90%. In caso di allarme, controlla i processi che consumano maggiormente la CPU. `ps -aux --sort=-%cpu | head -n 10`
- `CLUSTER_NAME-HeadNode-Mem`: rosso se l'utilizzo della memoria è superiore al 90%. In caso di allarme, controlla i processi che consumano maggiormente la memoria. `ps -aux --sort=-%mem | head -n 10`
- `CLUSTER_NAME-HeadNode-Disk`: rosso se lo spazio occupato su disco è superiore al 90% sul percorso `/`. In caso di allarme, controlla le cartelle che occupano la maggior parte dello spazio `condu -h --max-depth=2 / 2> /dev/null | sort -hr`.

Risoluzione delle modifiche alla configurazione del sistema operativo che causano errori o guasti

Quando si apportano modifiche alla configurazione del sistema operativo ai AWS ParallelCluster nodi, possono verificarsi vari problemi che possono causare errori nella creazione, nell'aggiornamento o nel funzionamento del cluster. Questa sezione fornisce indicazioni per identificare e risolvere i problemi più comuni relativi alla configurazione del sistema operativo.

Problemi comuni di configurazione del sistema operativo

Problemi di configurazione locale

Uno dei problemi di configurazione del sistema operativo più comuni è legato alle impostazioni locali. Se vedi errori come:

```
cannot change locale (en_US.utf-8) because it has an invalid name
```

Ciò si verifica in genere quando:

- Un processo yum di installazione non ha avuto successo e le impostazioni locali sono rimaste in uno stato incoerente

- Un utente ha interrotto prematuramente un processo di installazione
- I pacchetti locali sono mancanti o danneggiati

Come diagnosticare

1. Controlla se puoi passare all'utente `pcluster-admin`:

```
$ su - pcluster-admin
```

Se vedi un errore del tipo `cannot change locale...no such file or directory`, questo conferma il problema.

2. Controlla le lingue disponibili:

```
$ localedef --list
```

Se restituisce un elenco vuoto o non contiene le impostazioni locali predefinite, la configurazione locale non funziona.

3. Controlla l'ultimo yum comando:

```
$ yum history  
$ yum history info #ID
```

Se l'ultimo ID non è presente `Return-Code: Success`, gli script di post-installazione potrebbero non essere stati eseguiti correttamente.

Come risolvere

Ricostruisci la versione locale reinstallando i language pack:

```
$ sudo yum reinstall glibc-all-langpacks
```

Dopo la ricostruzione, verifica che il problema sia stato risolto eseguendo:

```
$ su - pcluster-admin
```

Se non viene visualizzato alcun errore o avviso, il problema è stato risolto.

Conflitti tra pacchetti OS

Quando si installano pacchetti personalizzati o si modificano pacchetti di sistema, possono verificarsi conflitti che impediscono il corretto funzionamento del cluster.

Come diagnosticare

1. Controlla il registro chef-client per gli errori relativi al pacchetto:

```
$ less /var/log/chef-client.log
```

2. Cerca i conflitti di dipendenza dei pacchetti nel log cfn-init:

```
$ less /var/log/cfn-init.log
```

Come risolvere

1. Se un pacchetto specifico causa problemi, prova a reinstallarlo:

```
$ sudo yum reinstall package-name
```

2. Per i conflitti di dipendenza, potrebbe essere necessario rimuovere i pacchetti in conflitto:

```
$ sudo yum remove conflicting-package
```

3. Se il problema persiste, prendi in considerazione la possibilità di creare un'AMI personalizzata con i pacchetti richiesti preinstallati utilizzando il `pcluster build-image` comando. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

Modifiche al file di configurazione del sistema

La modifica dei file di configurazione di sistema critici può causare errori del cluster, soprattutto se tali file sono gestiti da AWS ParallelCluster.

Come effettuare la diagnosi

1. Verifica la presenza di errori nel registro chef-client che menzionano file di configurazione specifici:

```
$ grep -i "config" /var/log/chef-client.log
```

2. Cerca errori di autorizzazione o sintassi nei file di configurazione:

```
$ less /var/log/cfn-init.log
```

Come risolvere

1. Ripristina i file di configurazione modificati allo stato originale:

```
$ sudo cp /etc/file.conf.bak /etc/file.conf
```

2. Se devi apportare modifiche persistenti ai file di configurazione del sistema, utilizza azioni di bootstrap personalizzate anziché modificare direttamente i file:

```
HeadNode:
  CustomActions:
    OnNodeConfigured:
      Script: s3://bucket-name/config-script.sh
```

Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).

3. Per le modifiche alla configurazione che devono essere apportate direttamente ai file di sistema, prendi in considerazione la creazione di un'AMI personalizzata. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

Aggiornamenti del kernel e problemi di compatibilità

Gli aggiornamenti del kernel possono causare problemi di compatibilità con determinati AWS servizi, in particolare con Amazon FSx for Lustre.

Come diagnosticare

1. Controlla se gli aggiornamenti del kernel sono stati applicati:

```
$ uname -r
```

2. Cerca gli errori di FSx montaggio di Amazon nei log:

```
$ grep -i "fsx" /var/log/chef-client.log
```

Come risolvere

1. Per Ubuntu 22.04, evita l'aggiornamento al kernel più recente in quanto non esiste un FSx client Amazon per quel kernel. Per ulteriori informazioni, consulta [Considerazioni sul sistema operativo](#).
2. Se hai già aggiornato il kernel e riscontri problemi, valuta la possibilità di effettuare il downgrade a una versione del kernel compatibile:

```
$ sudo apt install linux-image-previous-version
```

3. Per personalizzazioni persistenti del kernel, crea un'AMI personalizzata con la versione del kernel specifica di cui hai bisogno. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).

Le migliori pratiche per le modifiche alla configurazione del sistema operativo

Per ridurre al minimo i problemi relativi alle modifiche alla configurazione del sistema operativo:

1. Usa azioni Bootstrap personalizzate: invece di modificare direttamente i file di sistema, usa `OnNodeConfigured` gli script `OnNodeStart` o per apportare modifiche in modo controllato. Per ulteriori informazioni, consulta [Azioni bootstrap personalizzate](#).
2. Creazione personalizzata AMIs: per modifiche significative al sistema operativo, crea un'AMI personalizzata utilizzando `pcluster build-image` anziché apportare modifiche, alle istanze in esecuzione. Per ulteriori informazioni, consulta [AWS ParallelCluster Personalizzazione AMI](#).
3. Verifica prima le modifiche: prima di applicare le modifiche a un cluster di produzione, testale su un piccolo cluster di test per garantire la compatibilità.
4. Modifiche al documento: tieni traccia di tutte le modifiche alla configurazione del sistema operativo apportate per facilitare la risoluzione dei problemi.
5. File di configurazione di backup: prima di modificare qualsiasi file di configurazione del sistema, crea un backup:

```
$ sudo cp /etc/file.conf /etc/file.conf.bak
```

6. Controlla i log dopo le modifiche: dopo aver apportato modifiche alla configurazione del sistema operativo, controlla i log per eventuali errori:

```
$ less /var/log/cfn-init.log
```

```
$ less /var/log/chef-client.log
```

Seguendo queste linee guida, è possibile ridurre al minimo il rischio che le modifiche alla configurazione del sistema operativo causino errori del cluster e risolvere in modo più efficace eventuali problemi.

AWS ParallelCluster politica di supporto

AWS ParallelCluster supporta più versioni contemporaneamente. Ogni AWS ParallelCluster versione ha una data programmata per la fine del ciclo di vita del supporto (EOSL). Dopo la data EOSL, non viene fornito ulteriore supporto o manutenzione per quella versione.

AWS ParallelCluster utilizza uno schema di `major.minor.patch` versioni. Nuove funzionalità, miglioramenti delle prestazioni, aggiornamenti di sicurezza e correzioni di bug sono inclusi nelle nuove versioni secondarie relative all'ultima versione principale. Le versioni secondarie sono retrocompatibili all'interno di una versione principale. Per i problemi critici, AWS fornisce correzioni tramite rilasci di patch, ma solo per le ultime versioni secondarie delle versioni che non hanno raggiunto la EOSL. Se si desidera utilizzare gli aggiornamenti di una nuova versione, è necessario eseguire l'aggiornamento alla nuova versione secondaria o alla nuova versione patch.

AWS ParallelCluster versioni	Data di fine del ciclo di vita supportato (EOSL)
3.0. x	31/3/2023
3.1. x	8/31/2023
3.2. x	1/31/2024
3.3. x	5/31/2024
3.4. x	6/28/2024
3.5. x	8/31/2024
3.6. x	30/11/2024
3.7. x	2/28/2025
3.8. x	30/6/2025
3.9. x	09/05/2025
3.10. x	12/27/2025

AWS ParallelCluster versioni	Data di fine del ciclo di vita supportato (EOSL)
3.11. x	25/03/2026
3,12. x	30/06/2026
3,13. x	09/30/2026
3,14. x	29/03/2027

Sicurezza in AWS ParallelCluster

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te AWS e te. Il [modello di responsabilità condivisa](#) descrive questo approccio 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 ParallelCluster, consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio o dai servizi specifici che utilizzi. Sei inoltre responsabile di diversi altri fattori correlati, tra cui la sensibilità dei tuoi dati, i requisiti aziendali e le leggi e i regolamenti applicabili.

Questa documentazione descrive come applicare il modello di responsabilità condivisa durante l'utilizzo AWS ParallelCluster. I seguenti argomenti mostrano come configurare per AWS ParallelCluster soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare AWS ParallelCluster in modo da monitorare e proteggere AWS le tue risorse.

Argomenti

- [Informazioni di sicurezza per i servizi utilizzati da AWS ParallelCluster](#)
- [Protezione dei dati in AWS ParallelCluster](#)
- [Identity and Access Management per AWS ParallelCluster](#)
- [Convalida della conformità per AWS ParallelCluster](#)
- [Applicazione di una versione minima di TLS 1.2](#)
- [Configurazione di gruppi di sicurezza per ambienti con restrizioni](#)

Informazioni di sicurezza per i servizi utilizzati da AWS ParallelCluster

- [Sicurezza in Amazon EC2](#)
- [Sicurezza in Amazon API Gateway](#)
- [Sicurezza in AWS Batch](#)
- [Sicurezza in CloudFormation](#)
- [Sicurezza in Amazon CloudWatch](#)
- [Sicurezza in AWS CodeBuild](#)
- [Sicurezza in Amazon DynamoDB](#)
- [Sicurezza in Amazon ECR](#)
- [Sicurezza in Amazon ECS](#)
- [Sicurezza in Amazon EFS](#)
- [Sicurezza in FSx for Lustre](#)
- [Sicurezza in AWS Identity and Access Management \(IAM\)](#)
- [Sicurezza in EC2 Image Builder](#)
- [Sicurezza in AWS Lambda](#)
- [Sicurezza in Amazon Route 53](#)
- [Sicurezza in Amazon SNS](#)
- [Sicurezza in Amazon SQS \(per la AWS ParallelCluster versione 2.x.\)](#)
- [Sicurezza in Amazon S3](#)
- [Sicurezza in Amazon VPC](#)

Protezione dei dati in AWS ParallelCluster

Il [modello di](#) si applica alla protezione dei dati in. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il [General Data Protection Regulation \(GDPR\) Center](#).

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori o Servizi AWS utilizzi la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Crittografia dei dati

Una caratteristica fondamentale di qualsiasi servizio sicuro è che le informazioni vengano crittografate quando non sono utilizzate attivamente.

Crittografia dei dati a riposo

AWS ParallelCluster di per sé non memorizza dati del cliente diversi dalle credenziali necessarie per interagire con i AWS servizi per conto dell'utente.

Per i dati sui nodi del cluster, i dati possono essere crittografati quando sono inattivi.

Per i volumi Amazon EBS, la crittografia viene configurata utilizzando le KmsKeyId impostazioni [EbsSettingsEbsSettings/Encrypted](#) e nella [EbsSettings](#) sezione. Per ulteriori informazioni, consulta la [crittografia di Amazon EBS](#) nella Guida per l'utente di Amazon EC2.

Per i volumi Amazon EFS, la crittografia viene configurata utilizzando le KmsKeyId impostazioni [EfsSettingsEfsSettings/Encrypted](#) e nella [EfsSettings](#) sezione. Per ulteriori informazioni, consulta [How encryption at rest nella](#) Amazon Elastic File System User Guide.

Per i file system FSx for Lustre, la crittografia dei dati inattivi viene abilitata automaticamente durante la creazione di un file system Amazon FSx. Per ulteriori informazioni, [consulta Encrypting data at rest](#) nella Amazon FSx for Lustre User Guide.

Per i tipi di istanze con volumi NVMe, i dati sui volumi di NVMe Instance Store vengono crittografati utilizzando un XTS-AES-256 codice implementato su un modulo hardware sull'istanza. Le chiavi crittografiche sono generate utilizzando il modulo hardware e sono univoche per ciascun dispositivo di archivio NVMe. Quando l'istanza viene arrestata o terminata, tutte le chiavi crittografiche vengono distrutte e non possono essere ripristinate. Non è possibile disattivare questa cifratura e non è possibile fornire una propria chiave crittografica. Per ulteriori informazioni, consulta [Encryption at rest](#) nella Amazon EC2 User Guide.

Se utilizzi AWS ParallelCluster un AWS servizio che trasmette i dati dei clienti al computer locale per l'archiviazione, consulta il capitolo Sicurezza e conformità della Guida per l'utente del servizio per informazioni su come tali dati vengono archiviati, protetti e crittografati.

Crittografia dei dati in transito

Per impostazione predefinita, tutti i dati trasmessi dal computer client in esecuzione AWS ParallelCluster e dagli endpoint del AWS servizio vengono crittografati inviandoli tramite una connessione. HTTPS/TLS Il traffico tra i nodi del cluster può essere crittografato automaticamente, a seconda dei tipi di istanza selezionati. Per ulteriori informazioni, consulta [Encryption in transit](#) nella Amazon EC2 User Guide.

Consulta anche

- [Protezione dei dati in Amazon EC2](#)
- [Protezione dei dati in EC2 Image Builder](#)
- [Protezione dei dati in CloudFormation](#)
- [Protezione dei dati in Amazon EFS](#)
- [Protezione dei dati in Amazon S3](#)

- [Protezione dei dati in FSx for Lustre](#)

Identity and Access Management per AWS ParallelCluster

AWS ParallelCluster utilizza i ruoli per accedere alle AWS risorse e ai relativi servizi. Le politiche relative alle istanze e agli utenti AWS ParallelCluster utilizzate per concedere le autorizzazioni sono documentate in [AWS Identity and Access Management autorizzazioni in AWS ParallelCluster](#)

L'unica differenza principale è il modo in cui si esegue l'autenticazione quando si utilizza un utente standard e credenziali a lungo termine. Sebbene un utente richieda una password per accedere alla console di un AWS servizio, lo stesso utente richiede una coppia di key di accesso per eseguire le stesse operazioni utilizzando AWS ParallelCluster. Tutte le altre credenziali a breve termine vengono utilizzate nello stesso modo in cui vengono utilizzate con la console.

Le credenziali utilizzate da AWS ParallelCluster sono archiviate in file di testo semplice e non sono crittografate.

- Il `$HOME/.aws/credentials` file memorizza le credenziali a lungo termine necessarie per accedere alle risorse. AWS Per recuperare l'ID chiave di accesso e la chiave di accesso segreta
- Le credenziali a breve termine, ad esempio quelle per i ruoli assunti dall'utente o quelle relative AWS IAM Identity Center ai servizi, vengono inoltre archiviate rispettivamente nelle `$HOME/.aws/sso/cache` cartelle `$HOME/.aws/cli/cache` e.

Mitigazione del rischio

- Si consiglia vivamente di configurare le autorizzazioni del file system per la cartella `$HOME/.aws` e per le relative cartelle e file figlio per limitare l'accesso solo agli utenti autorizzati.
- Utilizzare i ruoli con credenziali temporanee laddove possibile per ridurre l'opportunità di danni in caso di compromissione delle credenziali. Utilizzare le credenziali a lungo termine solo per richiedere e aggiornare le credenziali del ruolo a breve termine.

Convalida della conformità per AWS ParallelCluster

I revisori esterni valutano la sicurezza e la conformità dei AWS servizi nell'ambito di più programmi di AWS conformità. L'utilizzo AWS ParallelCluster per accedere a un servizio non altera la conformità di tale servizio.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta la sezione [AWS Servizi rientranti nell'ambito dei programmi di conformità](#) (). Per informazioni generali, vedere programmi di [AWS conformità, programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando. AWS Artifact Per ulteriori informazioni, consulta [Download di report in AWS Artifact](#).

La vostra responsabilità di conformità durante l'utilizzo AWS ParallelCluster 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 su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- [Whitepaper sull'architettura per la sicurezza e la conformità HIPAA su Amazon Web Services: questo AWS white paper](#) descrive come le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- AWS risorse per [la conformità e risorse per la conformità](#): questa raccolta di potrebbe riguardare il tuo settore e la tua area geografica.
- [Valutazione delle risorse in base alle regole contenute](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub CSPM](#)— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Applicazione di una versione minima di TLS 1.2

Per aumentare la sicurezza durante la comunicazione con AWS i servizi, è necessario configurare l'utilizzo di TLS AWS ParallelCluster 1.2 o versione successiva. Quando si utilizza AWS ParallelCluster, Python viene utilizzato per impostare la versione TLS.

Per garantire che non AWS ParallelCluster utilizzi una versione TLS precedente a TLS 1.2, potrebbe essere necessario ricompilare OpenSSL per applicare questo valore minimo e quindi ricompilare Python per utilizzare OpenSSL di nuova generazione.

Determinare i protocolli attualmente supportati

Innanzitutto, creare un certificato autofirmato da utilizzare per il server di test e per l'SDK Python che utilizza OpenSSL.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Quindi avviare un server di test usando OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

In una nuova finestra del terminale, creare un ambiente virtuale e installare Python SDK.

```
$ python3 -m venv test-env  
source test-env/bin/activate  
pip install botocore
```

Creare un nuovo script Python denominato `check.py` che utilizza la libreria HTTP sottostante dell'SDK.

```
$ import urllib3  
URL = 'https://localhost:4433/'  
  
http = urllib3.PoolManager(  
    ca_certs='cert.pem',  
    cert_reqs='CERT_REQUIRED',  
)  
r = http.request('GET', URL)  
print(r.data.decode('utf-8'))
```

Eseguire il nuovo script.

```
$ python check.py
```

In questo modo vengono visualizzati i dettagli sulla connessione effettuata. Cercare "Protocol : " nell'output. Se l'output è "TLSv1.2" o successivo, l'SDK utilizza per impostazione predefinita TLS v1.2 o versione successiva. Se si tratta di una versione precedente, è necessario ricompilare OpenSSL e ricompilare Python.

Tuttavia, anche se l'installazione di Python è impostata su TLS v1.2 o versioni successive, è comunque possibile per Python rinegoziare una versione precedente a TLS v1.2 se il server non supporta TLS v1.2 o versioni successive. Per verificare che Python non rinegozi automaticamente su versioni precedenti, riavviare il server di test con quanto segue.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Se si sta utilizzando una versione precedente di OpenSSL, il flag `-no_tls1_3` potrebbe non essere disponibile. In questo caso, rimuovere il flag perché la versione di OpenSSL che si sta utilizzando non supporta TLS v1.3. Eseguire lo script Python.

```
$ python check.py
```

Se l'installazione di Python non rinegozia correttamente per le versioni precedenti a TLS 1.2, si dovrebbe ricevere un errore SSL.

```
$ urllib3.exceptions.MaxRetryError: HTTPConnectionPool(host='localhost',  
port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL:  
UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

Se si è in grado di stabilire una connessione, ricompilare OpenSSL e Python per disabilitare la negoziazione dei protocolli precedenti a TLS v1.2.

Compilare OpenSSL e Python

Per assicurarti che AWS ParallelCluster ciò non valga per nulla precedente a TLS 1.2, devi ricompilare OpenSSL e Python. A tale scopo, copiare il contenuto seguente per creare uno script ed eseguirlo.

```
#!/usr/bin/env bash  
set -e  
  
OPENSSL_VERSION="1.1.1d"  
OPENSSL_PREFIX="/opt/openssl-with-min-tls1_2"  
PYTHON_VERSION="3.8.1"  
PYTHON_PREFIX="/opt/python-with-min-tls1_2"  
  
curl -O "https://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz"
```

```
tar -xzf "openssl-$OPENSSL_VERSION.tar.gz"
cd openssl-$OPENSSL_VERSION
./config --prefix=$OPENSSL_PREFIX no-ssl3 no-tls1 no-tls1_1 no-shared
make > /dev/null
sudo make install_sw > /dev/null

cd /tmp
curl -O "https://www.python.org/ftp/python/$PYTHON_VERSION/Python-$PYTHON_VERSION.tgz"
tar -xzf "Python-$PYTHON_VERSION.tgz"
cd Python-$PYTHON_VERSION
./configure --prefix=$PYTHON_PREFIX --with-openssl=$OPENSSL_PREFIX --disable-shared > /dev/null
make > /dev/null
sudo make install > /dev/null
```

Questa operazione esegue la compilazione di una versione di Python che ha un OpenSSL collegato staticamente che non negozia automaticamente versioni precedenti a TLS 1.2. Questa operazione installa anche OpenSSL nella directory `/opt/openssl-with-min-tls1_2` e installa Python nella directory `/opt/python-with-min-tls1_2`. Dopo aver eseguito questo script, confermare l'installazione della nuova versione di Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Questa operazione dovrebbe stampare quanto segue.

```
Python 3.8.1
```

Per confermare che questa nuova versione di Python non negozi una versione precedente a TLS 1.2, eseguire nuovamente le fasi a partire da [Determinare i protocolli attualmente supportati](#) utilizzando la versione di Python appena installata (ovvero `/opt/python-with-min-tls1_2/bin/python3`).

Configurazione di gruppi di sicurezza per ambienti con restrizioni

Per impostazione predefinita, AWS ParallelCluster crea e configura gruppi di sicurezza che consentono tutto il traffico tra i nodi del cluster. In ambienti altamente limitati, potrebbe essere necessario limitare l'accesso alla rete solo alle porte necessarie per il funzionamento del cluster. Questa sezione descrive come configurare gruppi di sicurezza personalizzati con accesso limitato per la AWS ParallelCluster distribuzione.

Panoramica dei gruppi di sicurezza

AWS ParallelCluster utilizza i gruppi di sicurezza per controllare il traffico di rete tra il nodo principale, i nodi di elaborazione e i nodi di accesso (se configurati). Per impostazione predefinita, quando AWS ParallelCluster crea un cluster, crea gruppi di sicurezza che consentono tutto il traffico tra i nodi all'interno del cluster. In ambienti con requisiti di sicurezza rigorosi, è possibile fornire gruppi di sicurezza personalizzati che limitano il traffico solo alle porte necessarie.

I gruppi di sicurezza possono essere configurati nelle seguenti sezioni della configurazione del cluster:

- [HeadNode/Networking](#)- Controlla l'accesso da e verso il nodo principale
- [Scheduling/SlurmQueues/Networking](#)- Controlla l'accesso da e verso i nodi di calcolo
- [LoginNodes](#)- Controlla l'accesso da e verso i nodi di accesso (se configurato)

Per ognuna di queste sezioni, puoi specificare:

- `SecurityGroups`- Sostituisce i gruppi di sicurezza predefiniti che AWS ParallelCluster verrebbero creati
- `AdditionalSecurityGroups`- Aggiunge gruppi di sicurezza oltre a quelli predefiniti creati da AWS ParallelCluster

Porte necessarie per il funzionamento del cluster

Quando si configurano gruppi di sicurezza personalizzati, è necessario assicurarsi che le seguenti porte siano aperte tra i nodi appropriati:

Porte richieste per il nodo principale

Porta	Protocollo	Direzione	Scopo
22	TCP	In entrata	Accesso SSH al nodo principale (dagli intervalli IP consentiti)
6817-6819	TCP	In entrata	Porte del controller Slurm (dai nodi di calcolo e di accesso)

Porta	Protocollo	Direzione	Scopo
6817-6819	TCP	In uscita	Porte del controller Slurm (per calcolare e accedere ai nodi)
8443	TCP	In entrata	NICE DCV (se abilitato, dagli intervalli IP consentiti)
11, 2049	TCP/UDP	In entrata	NFS (dai nodi di calcolo e di accesso, se si utilizza NFS per lo storage condiviso)
443	TCP	In uscita	Accesso HTTPS ai AWS servizi (se non si utilizzano endpoint VPC)

Porte richieste per i nodi di calcolo

Porta	Protocollo	Direzione	Scopo
22	TCP	In entrata	Accesso SSH (dal nodo principale e dai nodi di accesso)
6818	TCP	In entrata	porta daemon Slurm (dal nodo principale)
6817-6819	TCP	In uscita	Porte del controller Slurm (verso il nodo principale)
111, 2049	TCP/UDP	In uscita	NFS (al nodo principale, se si utilizza NFS per lo storage condiviso)
443	TCP	In uscita	Accesso HTTPS ai AWS servizi (se non si utilizzano endpoint VPC)

Se utilizzi EFA (Elastic Fabric Adapter), devi inoltre consentire tutto il traffico tra i nodi di elaborazione che hanno EFA abilitato:

- Tutto il traffico TCP e UDP tra i nodi di elaborazione con EFA

- Tutto il traffico sul dispositivo EFA tra i nodi di elaborazione con EFA

Note

Se utilizzi sistemi di storage condivisi come FSx Lustre, Amazon EFS o altre soluzioni di storage, dovrai assicurarti che le porte appropriate siano aperte anche per questi servizi.

Creazione di gruppi di sicurezza personalizzati

Per creare gruppi di sicurezza personalizzati per la tua AWS ParallelCluster distribuzione, segui questi passaggi:

1. Crea gruppi di sicurezza per il nodo principale, i nodi di calcolo e i nodi di accesso (se applicabile) utilizzando la console di AWS gestione, la AWS CLI o AWS CloudFormation
2. Configura le regole del gruppo di sicurezza per consentire solo il traffico necessario, come indicato nella sezione precedente.
3. Fai riferimento a questi gruppi di sicurezza nel file di configurazione del cluster.

Ecco un esempio di come creare gruppi di sicurezza utilizzando la AWS CLI:

```
# Create security group for head node
aws ec2 create-security-group \
  --group-name pcluster-head-node-sg \
  --description "Security group for ParallelCluster head node" \
  --vpc-id vpc-12345678

# Create security group for compute nodes
aws ec2 create-security-group \
  --group-name pcluster-compute-node-sg \
  --description "Security group for ParallelCluster compute nodes" \
  --vpc-id vpc-12345678

# Add rules to allow necessary traffic between head and compute nodes
# (Add specific rules based on the required ports listed above)
```

Configurazione dei gruppi di sicurezza nella configurazione del cluster

Dopo aver creato i gruppi di sicurezza personalizzati, puoi farvi riferimento nel file di configurazione del cluster:

```
# Example cluster configuration with custom security groups
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678
    SecurityGroups:
      - sg-headnode12345 # Custom security group for head node
      # Or use AdditionalSecurityGroups if you want to keep the default security groups
      # AdditionalSecurityGroups:
      #   - sg-additional12345
    ...

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
    Networking:
      SubnetIds:
        - subnet-12345678
      SecurityGroups:
        - sg-computenode12345 # Custom security group for compute nodes
        # Or use AdditionalSecurityGroups if you want to keep the default security
groups
        # AdditionalSecurityGroups:
        #   - sg-additional12345
    ...

# If using login nodes
LoginNodes:
  Pools:
    - Name: login-pool
    ...
    Networking:
      SubnetIds:
        - subnet-12345678
      SecurityGroups:
        - sg-loginnode12345 # Custom security group for login nodes
```

```
# Or use AdditionalSecurityGroups if you want to keep the default security
groups
# AdditionalSecurityGroups:
#   - sg-additional12345
...
```

Durante l'utilizzo `SecurityGroups`, AWS ParallelCluster utilizzerà solo i gruppi di sicurezza specificati, sostituendo quelli predefiniti. Durante l'utilizzo `AdditionalSecurityGroups`, AWS ParallelCluster utilizzerà sia i gruppi di sicurezza predefiniti che crea sia quelli aggiuntivi specificati dall'utente.

Warning

Se abiliti [Elastic Fabric Adapter \(EFA\)](#) per le tue istanze di calcolo, assicurati che le istanze abilitate per EFA siano membri di un gruppo di sicurezza che consente tutto il traffico in entrata e in uscita verso se stesso. Ciò è necessario per il corretto funzionamento di EFA.

Utilizzo degli endpoint VPC in ambienti con restrizioni

In ambienti altamente limitati, potresti voler eseguire l'implementazione AWS ParallelCluster in una sottorete senza accesso a Internet. In questo caso, dovrai configurare gli endpoint VPC per consentire al cluster di comunicare con i servizi. AWS Per istruzioni dettagliate, vedi [AWS ParallelCluster in un'unica sottorete senza accesso a Internet](#).

Quando utilizzi gli endpoint VPC, assicurati che i tuoi gruppi di sicurezza consentano il traffico da e verso gli endpoint VPC. Puoi farlo aggiungendo i gruppi di sicurezza associati agli endpoint VPC alla `AdditionalSecurityGroups` configurazione del nodo principale e dei nodi di calcolo.

```
HeadNode:
...
Networking:
  SubnetId: subnet-1234567890abcdef0
  AdditionalSecurityGroups:
    - sg-abcdef01234567890 # Security group that enables communication with VPC
    endpoints
  ...

Scheduling:
  Scheduler: slurm
```

```
SlurmQueues:
- ...
  Networking:
    SubnetIds:
      - subnet-1234567890abcdef0
    AdditionalSecurityGroups:
      - sg-1abcdef01234567890 # Security group that enables communication with VPC
endpoints
```

Le migliori pratiche per la configurazione dei gruppi di sicurezza

Quando configuri i gruppi di sicurezza per ambienti con restrizioni, prendi AWS ParallelCluster in considerazione le seguenti best practice:

- Principio del privilegio minimo: aprire solo le porte necessarie per il funzionamento del cluster.
- Utilizza i riferimenti ai gruppi di sicurezza: quando possibile, utilizza i riferimenti ai gruppi di sicurezza (che consentono il traffico proveniente da un altro gruppo di sicurezza) anziché i blocchi CIDR per limitare il traffico tra i componenti del cluster.
- Limita l'accesso SSH: limita l'accesso SSH al nodo principale solo agli intervalli IP che lo richiedono utilizzando la configurazione [HeadNode//Ssh](#). `AllowedIps`
- Limita l'accesso a DCV: se usi NICE DCV, limita l'accesso solo agli intervalli IP che lo richiedono utilizzando [HeadNode della configurazione Dcv//AllowedIps](#).
- Esegui test approfonditi: dopo aver configurato i gruppi di sicurezza personalizzati, verifica accuratamente tutte le funzionalità del cluster per assicurarti che tutti i percorsi di comunicazione richiesti funzionino.
- Documenta la tua configurazione: conserva la documentazione della configurazione del tuo gruppo di sicurezza, comprese le porte aperte e il motivo per cui sono necessarie.

Risoluzione dei problemi relativi ai gruppi di sicurezza

Se riscontri problemi dopo la configurazione dei gruppi di sicurezza personalizzati, considera i seguenti passaggi per la risoluzione dei problemi:

- Controlla i log del cluster: esamina i log del cluster in CloudWatch Logs per eventuali errori di connessione.
- Verifica le regole del gruppo di sicurezza: assicurati che tutte le porte richieste siano aperte tra i nodi appropriati.

- Verifica la connettività: utilizza strumenti come `telnet` o `nc` per testare la connettività tra nodi su porte specifiche.
- Espandi temporaneamente le regole: se hai problemi a identificare le porte necessarie, consenti temporaneamente tutto il traffico tra i nodi del cluster e poi limitalo gradualmente man mano che identifichi le porte richieste.
- Verifica la configurazione degli endpoint VPC: se utilizzi gli endpoint VPC, assicurati che siano configurati correttamente e che i gruppi di sicurezza consentano il traffico da e verso di essi.

Se continui a riscontrare problemi, puoi tornare a utilizzare i gruppi di sicurezza predefiniti creati rimuovendo la configurazione AWS ParallelCluster dal file di `SecurityGroups` configurazione del cluster.

Supportato Regioni AWS per AWS ParallelCluster

AWS ParallelCluster la versione 3 è disponibile nelle seguenti versioni Regioni AWS:

Nome della regione	Regione	Prima AWS ParallelCluster versione che supporta la regione
Stati Uniti orientali (Ohio)	us-east-2	3.0.0
US East (N. Virginia)	us-east-1	3.0.0
US West (N. California)	us-west-1	3.0.0
US West (Oregon)	us-west-2	3.0.0
Africa (Cape Town)	af-south-1	3.0.0
Asia Pacifico (Hong Kong)	ap-east-1	3.0.0
Asia Pacific (Mumbai)	ap-south-1	3.0.0
Asia Pacifico (Seul)	ap-northeast-2	3.0.0
Asia Pacific (Singapore)	ap-southeast-1	3.0.0
Asia Pacific (Sydney)	ap-southeast-2	3.0.0
Asia Pacifico (Giacarta)	ap-southeast-3	3.10.0
Asia Pacifico (Malesia)	ap-southeast-5	3,13,0
Asia Pacifico (Tailandia)	ap-southeast-7	3.13.0
Asia Pacifico (Tokyo)	ap-northeast-1	3.0.0
Canada (Central)	ca-central-1	3.0.0
China (Beijing)	cn-north-1	3.0.0
Cina (Ningxia)	cn-northwest-1	3.0.0

Nome della regione	Regione	Prima AWS ParallelCluster versione che supporta la regione
Europe (Frankfurt)	eu-central-1	3.0.0
Europa (Irlanda)	eu-west-1	3.0.0
Europe (London)	eu-west-2	3.0.0
Europa (Milano)	eu-south-1	3.0.0
Europe (Paris)	eu-west-3	3.0.0
Europa (Stoccolma)	eu-north-1	3.0.0
Medio Oriente (Bahrein)	me-south-1	3.0.0
Sud America (São Paulo)	sa-east-1	3.0.0
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	3.0.0
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	3.0.0
Israele (Tel Aviv)	il-central-1	3.8.0

Note di rilascio e cronologia dei documenti

Le tabelle seguenti descrivono i principali aggiornamenti AWS ParallelCluster e le nuove funzionalità della Guida per l'utente. Inoltre, aggiorniamo frequentemente la documentazione per tener conto del feedback inviatoci.

AWS ParallelCluster

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.15.0	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Miglioramenti: • Aggiungi il supporto per le istanze p6-b300 per tutti i sistemi operativi tranne AL2. • Sostituisci cfn-hup nei nodi di calcolo con systemd timer per supportare gli aggiornamenti in loco al fine di migliorare le prestazioni per carichi di lavoro strettamente collegati su larga scala. • Disabilita <code>dnf-makecache.timer</code> per migliorare le prestazioni per carichi di lavoro strettamente collegati su larga scala. RHEL/Rocky • Supporta gli aggiornamenti Tags durante gli aggiornamenti del cluster.	23 marzo 2026

Modifica	Descrizione	Data
	• Aggiungi <code>LaunchTemplateOverrides</code> alla configurazione del cluster per consentire la personalizzazione delle interfacce di rete sovrascrivendo il modello di avvio di una risorsa di calcolo. • Aggiungi un allarme relativo al battito cardiaco di <code>clustermgtd</code> mancante. Modifiche: • Quando EFA è abilitato, AWS ParallelCluster ora configura le interfacce di rete come <code>interface</code> e <code>efa-only</code> al posto del tipo combinato. <code>efa</code> Ciò riduce il consumo di indirizzi IP da uno per scheda di rete a uno per istanza. • Aggiorna Slurm alla versione 25.11.4 (dalla 24.11.7). • Aggiorna Pmix a 5.0.10 (dalla 5.0.6). • Aggiorna EFA installer a 1.47.0 (da 1.44.0). • Aggiorna il driver NVIDIA alla versione 580.105.08 (da 570.172.08) per tutti	

Modifica	Descrizione	Data
	i sistemi operativi tranne Amazon Linux 2. • Aggiorna GDRCopy alla versione 2.5.2 (dalla 2.4.4). • Aggiorna DCV alla versione 2025.0-20103 (dal 2024.0-19030). • Aggiorna CUDA Toolkit alla versione 13.0.2 (dalla 12.8.1) per tutti i sistemi operativi tranne Amazon Linux 2. • Aggiorna Python alla versione 3.14.2 (dalla versione 3.12.11) per tutti i sistemi operativi tranne Amazon Linux 2. • Aggiorna Cinc Client alla versione 18.8.54 (dalla 18.7.10). • Aggiorna Intel MPI Library alla versione 2021.17.2 (dalla 2021.16.0). • Aggiorna DCGM alla versione 4.5.1 (dalla 4.4.1) per tutti i sistemi operativi tranne Amazon Linux 2. • Riduci gli errori temporanei dell'immagine di compilazione in RHEL e Rocky causati da mirror di repository non sincronizzati	

Modifica	Descrizione	Data
	reimpostando i metadati al nuovo tentativo. • Avvia sempre <code>clustermgtd</code> in caso di aggiornamento del cluster e calcola l'errore di aggiornamento dello stato della flotta, indipendentemente dalla condizione di errore. • Migliora la resilienza del flusso di lavoro di rollback degli aggiornamenti del cluster. • Aggiorna NVIDIA Fabric Manager a 580.105.08 per tutti i sistemi operativi tranne Amazon Linux 2. • Aggiorna <code>aws-cfn-bootstrap</code> alla versione 2.0-38 (dalla 2.0-33). • Aggiorna <code>mysql-community-client</code> alla versione 8.4.8 (dalla 8.0.39) per tutti i sistemi operativi tranne Amazon Linux 2. • Aggiorna <code>amazon-efs-utils</code> alla versione 2.4.0 (dalla v2.1.0) per le AMI Amazon Linux. • Aggiorna <code>jmespath</code> a <code>~=1.0</code> (da <code>~=0.10</code>). • Aggiorna <code>tabulate</code> a <code><=0.9.0</code> (da <code><=0.8.10</code>).	

Modifica	Descrizione	Data
	• Aggiungi un validatore per avvisare nel caso in cui gli aggiornamenti in loco siano stati disabilitati (tramite) sui nodi di calcolo e di accesso. DevSettings Correzioni di bug: • Risolvi il problema che LoginNodes NLB non era accessibile al pubblico quando si utilizza una sottorete pubblica con associazione implicita alla tabella di rotta principale. • Risolve un errore durante la creazione di un cluster con istanze GPU e con DCV abilitato ma senza accesso a Internet. • Risolve un problema per cui la creazione del cluster avrebbe avuto esito negativo a intermittenza a causa dell'eventuale coerenza quando i nodi head/compute /login condividono lo stesso gruppo di sicurezza. • Risolve l'errore di build-image durante l'installazione di ubuntu-desktop su un'immagine principale	

Modifica	Descrizione	Data
	di Ubuntu con pacchetti del sistema operativo obsoleti. • <code>HeadNode/LocalStorage</code> Correggi la convalida di. Questo parametro di configurazione non supporta gli aggiornamenti. • Correggi il validatore <code>PlacementGroupCapacityReservationValidator</code> per accettare prenotazioni di capacità con gruppi di collocamento tra più account. • Correggi la configurazione <code>CloudWatch</code> dell'agente per garantire la corretta analisi dei timestamp su tutti i file di registro. • Correggi la configurazione della registrazione per acquisire tutti gli eventi di controllo dello stato di salute di Slurm (aggiornamento del livello di registro da <code>WARNING</code> a <code>INFO</code> per evitare voci di registro mancanti). • Migliora la resilienza degli aggiornamenti del cluster assicurando che l'aggiornamento non abbia esito negativo sui nodi che	

Modifica	Descrizione	Data
	completano il bootstrap durante l'aggiornamento. • Impedisci l'esecuzione del processo di ripristino degli errori di aggiornamento del cluster sui cluster AWS Batch. Questo meccanismo di ripristino deve essere eseguito solo su cluster Slurm. Deprecazioni: • Il parametro <code>LoginNodes/Pool/s/Ssh/KeyName</code> di configurazione, obsoleto dalla versione 3.14.0, non è più supportato. • Questa è l'ultima AWS ParallelCluster versione che supporta Amazon Linux 2, poiché Amazon Linux 2 terminerà il supporto il 30 giugno 2026. • Questa è l'ultima AWS ParallelCluster versione che supporta AWS Batch CLI. A partire dalla v3.16.0, non AWS ParallelCluster supporterà più AWS Batch come scheduler.	

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.14.2	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Modifiche: • Aggiorna munge alla versione 0.5.18 (dalla 0.5.16) all'indirizzo. CVE-2026-25506 • Aggiorna la versione di NodeJS nel programma di installazione alla versione 22.22.0 (dalla 20.18.3).	16 febbraio 2026

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.14.1	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Modifiche: • Migliora la resilienza degli aggiornamenti del cluster assicurando che <code>clustermgtd</code> venga avviato dopo che gli aggiornamenti sono stati completati correttamente o dopo aggiornamenti non riusciti se la riconfigurazione della coda è riuscita. • Aggiungi l'attributo <code>chef cluster/in_place_update_on_fleet_enabled</code> per disabilitare gli aggiornamenti in loco sui nodi di calcolo e di accesso e mitigare l'impatto sulle prestazioni su larga scala. • Aggiorna Slurm alla versione 24.11.7 (dalla 24.11.6). • Aggiorna Werkzeug a <code>~3.1</code> (da <code>~2.0</code>) a <code>address</code>. CVE-2024-34069 • Aggiorna Connexion a <code>~2.15.1</code> (da <code>~2.13.0</code>). • Aggiorna Flask a <code>~3.1.0</code> (da <code>>=2.2.5, <2.3</code>).	22 dicembre 2025

Modifica	Descrizione	Data
	• Carica il modulo del kernel <code>drm_client_lib</code> prima dell'installazione del driver NVIDIA, se disponibile sul kernel. • Riduci l'impatto sulla dipendenza installando il pacchetto anziché. <code>sssd-common sssd</code> • Aggiorna <code>libjwt</code> alla versione 1.18.4 (dalla 1.17.0) per tutti i sistemi operativi tranne Amazon Linux 2. • Aggiorna <code>amazon-efs-utils</code> alla versione 2.4.0 (dalla v2.3.1). • Aggiorna EFA installer alla versione 1.44.0 (dalla 1.43.2). • Efa-driver: <code>efa-2.17.3-1</code> • Efa-config: <code>efa-config-1.18-1</code> • Efa-profile: <code>efa-profile-1.7-1</code> • Libfabric-aws: <code>libfabric-aws-2.3.1-1</code> • Rdma-core: <code>rdma-core-59.0-1</code> • Apri MPI: <code>openmpi40-aws-4.1.7-2</code> e <code>openmpi50-aws-5.0.8-11</code>	

Modifica	Descrizione	Data
	Correzioni di bug: • Risolve un problema per cui cfn-hup inseriva un loop infinito sul nodo principal e dopo un rollback a uno stato del cluster più vecchio di 24 ore, causato dal fatto che cfn-signal non riusciva a segnalare un handle di condizione di attesa scaduto. • Risolve la condizione di gara in cui i nodi di calcolo potevano implementare la versione di configurazione del cluster errata dopo un errore di aggiornamento. • Previene gli errori nei controlli di idoneità del cluster dovuti alle istanze avviate mentre il controllo è in corso. • Corregge l'analisi errata del timestamp per chef-client.log nella configurazione dell'agente. CloudWatch • Disabilita l'aggiornamento automatico di snap su Ubuntu durante la compilazione dell'immagine per evitare errori di riavvio intermittenti.	

Modifica	Descrizione	Data
	• Riduci il tempo di installazione di EFA per Ubuntu di ~20 minuti conservando solo i pacchetti del kernel per il kernel installato. • Aggiungi <code>GetFunction</code> e <code>GetPolicy</code> autorizzazioni per prevenire <code>AccessDenied</code> errori durante l'eliminazione dello <code>ParallelClusterBuildImageCleanupRole</code> stack di immagini di compilazione. • Correggi i messaggi di errore di convalida quando <code>DevSettings</code> è nullo o <code>DevSettings/InstanceTypesData</code> mancano i campi obbligatori.	

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.14.0	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Miglioramenti: • Includi driver P6e-GB200 e P6-B200 istanze. ParallelCluster configura un plugin di topologia Slurm da gestire. P6e-GB200 UltraServers Vedi la sezione Limitazioni per importanti requisiti di configurazione aggiuntivi. • Strategia di supporto prioritized e capacity-optimized-prioritized allocazione. Ciò consente agli utenti di dare priorità alle sottoreti, ad esempio il posizionamento per ottimizzare costi e prestazioni. • Aggiungi build-image il supporto per le AMI Amazon Linux 2023 basate sul kernel 6.12 (oltre alla versione 6.1). • Supporta DCV su Amazon Linux 2023. • Echo chef-client accede alla console dell'istanza quando un nodo non riesce	30 settembre 2025

Modifica	Descrizione	Data
	ad avviarsi. Questo aiuta a indagare sugli errori di bootstrap nei casi in cui i log non siano disponibili. CloudWatch Restrizioni: • P6e-GB200 le istanze vengono testate solo su Amazon Linux 2023, Ubuntu 22.04 e Ubuntu 24.04. • L'utilizzo di IMEX on P6e-GB200 richiede una configurazione aggiuntiva. Fai riferimento al tutorial dedicato nella nostra documentazione pubblica. • P6-B200 le istanze sono testate solo su Amazon Linux 2023, RHEL 8 e 9, Rocky 8 e 9, Ubuntu 22.04 e Ubuntu 24.04. • HealthChecks Le GPU non sono consigliate per le istanze con memoria GPU superiore a 320 GB (come p6-b200.48xlarge). La durata dell'Health Check può superare i 10 minuti, causando potenzialmente insuccessi e riducendo significativamente la produttività del lavoro.	

Modifica	Descrizione	Data
	Modifiche: • Installa nvidia-imex per tutti i sistemi operativi tranne Amazon Linux 2. • Rimuovi <code>UnkillableStepTimeout</code> da <code>slurm.conf</code> e lascia che slurm imposti questo valore. • Aggiorna il runtime Python usato dalle funzioni Lambda a Python 3.12 (da 3.9). Consulta la documentazione Lambda per informazioni importanti su Python 3.9 EOL. • Supporta la crittografia del file system EFS utilizzato per lo storage condiviso interno del nodo principale tramite un nuovo parametro di configurazione <code>HeadNode/SharedStorageEfsSettings/Encrypted</code>. • Aggiungi un validatore che avverte di non utilizzare istanze non GPU con DCV. • Aggiorna Slurm alla versione 24.11.6 (dalla 24.05.8). • Aggiorna EFA installer alla versione 1.43.2 (dalla 1.41.0).	

Modifica	Descrizione	Data
	- Efa-driver: efa-2.17.2-1 - Efa-config: efa-config-1.18-1 - Efa-profile: efa-profile-1.7-1 - Libfabric-aws: libfabric-aws-2.1.0-5 - Rdma-core: rdma-core-58.0-1 - Apri MPI: openmpi40-aws-4.1.7-2 e openmpi50-aws-5.0.6-11 - Aggiorna Cinc Client alla versione 18.4.12 (dalla 18.2.7). - Aggiorna il driver NVIDIA alla versione 570.172.08 (da 570.86.15) per tutti i sistemi operativi tranne Amazon Linux 2. - Aggiorna CUDA Toolkit alla versione 12.8.1 (dalla 12.8.0) per tutti i sistemi operativi tranne Amazon Linux 2. - Aggiorna DCGM alla versione 4.4.1 (dalla 3.3.6) per tutti i sistemi operativi tranne Amazon Linux 2. - Aggiorna Python alla versione 3.12.11 (dalla 3.12.8) per tutti i sistemi	

Modifica	Descrizione	Data
	operativi tranne Amazon Linux 2. • Aggiorna Python alla versione 3.9.23 (dalla 3.9.20) per Amazon Linux 2. • Aggiorna la libreria Intel MPI alla versione 2021.16.0 (dalla 2021.13.1). • Aggiorna DCV alla versione 2024.0-19030. • Aggiorna le AMI ufficiali di ParallelCluster Amazon Linux 2023 al kernel 6.12 (dalla versione 6.1). Correzioni di bug: • Previene gli errori di eliminazione <code>build-image</code> dello stack implementando un ruolo globale che elimina automaticamente lo <code>build-image</code> stack dopo che le immagini hanno avuto esito positivo o negativo nella compilazione. Il ruolo è destinato a esistere anche dopo l'eliminazione dello stack. Vedi https://github.com/aws/aws-parallelcluster/issues/5914. • Risolve un problema per cui la convalida del	

Modifica	Descrizione	Data
	gruppo di sicurezza non riusciva quando una regola conteneva sia intervalli IPv4 (IpRanges) che riferimenti al gruppo di sicurezza (). UserIdGroupPairs • <code>build-image</code> Risolve l'errore su Rocky 9, che si verificava quando l'immagine principale non contiene l'ultima versione del kernel sull'ultima versione secondaria di Rocky. • Risolve il problema di mancata corrispondenza degli ID del cluster che causa errori di aggiornamento del cluster quando viene utilizzato lo slurm accounting. • Risolve una condizione di gara all'avvio CloudWatch dell'agente che poteva causare errori nel bootstrap del nodo. Deprecazioni: • Il parametro di configurazione <code>LoginNodes/</code> <code>Pools/Ssh/KeyName</code> è obsoleto e verrà rimosso nelle versioni future. La CLI ora restituisce	

Modifica	Descrizione	Data
	ce un messaggio di avviso quando viene utilizzata nella configurazione del cluster. Vedi https://github.com/aws/aws-parallelcluster/issues/6811. • Ubuntu 20.04 non è più supportato.	
AWS ParallelCluster rilasciata la versione 3.13.2	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Correzioni di bug • Risolve un bug che può causare <code>update-cluster</code> e fallire quando le risorse di calcolo fanno riferimento <code>update-compute-fleet</code> a una riserva di capacità scaduta che non è più accessibile tramite le API EC2. • <code>build-image</code> Risolve l'errore su Rocky 9 che si verifica quando l'immagine principale non fornisce l'ultima versione del kernel. Vedi. https://github.com/aws/aws-parallelcluster/issues/6874	24 giugno 2025

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.13.1	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Modifiche: • Aggiorna Slurm alla versione 24.05.8. • Aggiorna il programma di installazione EFA alla versione 1.41.0 (dalla versione 1.38.1). • Efa-driver: efa-2.15.0-1 • Efa-config: efa-config-1.18-1 • Efa-profile: efa-profile-1.7-1 • Libfabric-aws: libfabric-aws-2.1.0-1 • Rdma-core: rdma-core-57.0-1 • Apri MPI: openmpi40-aws-4.1.7-2 e openmpi50-aws-5.0.6 • Aggiorna amazon-efs-utils alla versione 2.3.1 (dalla v2.1.0) per AMI non Amazon Linux. • Supporta DCV in us-isob-east-1 e us-iso-east-1.	04 giugno 2025

Modifica	Descrizione	Data
	• Supporta FSX per Lustre e Ontap in us-isob-east-1 e us-iso-east-1. • Garantisci la coerenza del kernel durante la creazione dell'immagine bloccando all'inizio e sbloccando al termine. ParallelCluster Correzioni di bug • Risolve un bug nell'installazione di ARM Performance Library che causava il fallimento dell'immagine di compilazione nelle regioni AWS Top Secret e Secret. AWS • Correzione di un bug che impediva allo script 'update_directory_service_password.sh' di aggiornare la password di AD.	

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.13.0	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Deprecazioni: - Questa è l'ultima ParallelCluster versione che supporta Ubuntu 20.04, poiché Ubuntu 20.04 sarà disponibile a maggio 2025. End-Of-Standard-Support Miglioramenti: - Aggiungi il supporto per Ubuntu 24.04. - Aggiungi il supporto per la regione ap-southeast-7. - Disattiva i service cup e wpa_suppllicant non utilizzati dalle AMI ufficiali per migliorare la sicurezza. ParallelCluster Modifiche: - Aggiorna Slurm alla versione 24.05.7. - Aggiorna il driver NVIDIA alla versione 570.86.15 (da 550.127.08) per tutti i sistemi operativi tranne AL2.	01 aprile 2025

Modifica	Descrizione	Data
	• Aggiorna CUDA Toolkit alla versione 12.8.0 (dalla 12.4.1) per tutti i sistemi operativi tranne AL2. • Aggiorna Python alla versione 3.12.8 per tutti i sistemi operativi tranne AL2 (dalla 3.9.20). • Su Ubuntu 22.04, installa il driver Nvidia con la stessa versione del compilatore utilizzata per compilare il kernel. • Aggiornamento alla versione 2.0-33. aws-cfn-bootstrap • Aggiorna il programma di installazione EFA alla versione 1.38.0 (dalla versione 1.36.0). • Efa-driver: efa-2.13.0-1 • Efa-config: efa-config-1.17-1 • Efa-profile: efa-profile-1.7-1 • Libfabric-aws : libfabric-aws-1.22.0-1 • Rdma-core: rdma-core-54.0-1 • Open MPI: e openmpi40-aws-4.1.7-1	

Modifica	Descrizione	Data
	openmpi50-aws-5.0.5 • Aggiorna amazon-efs-utils alla versione 2.1.0. • Rimuovi il ricettario di terze parti: apt-7.5.22 e pyenv-4.2.3. • Aggiorna le dipendenze dei libri di cucina di terze parti: • line-4.5.21 (dalla riga 4.5.13) • nfs-5.1.5 (da nfs-5.1.2) • openssh-2.11.14 (da openssh-2.11.12) • yum-7.4.20 (da yum-7.4.13) • yum-epel-5.0.8 (da yum-epel-5.0.2) • Aggiorna Pmix a 5.0.6 (da 5.0.3). • Aggiorna ARM PL alla versione 24.10 (dalla 23.10). • Aggiorna Python alla versione 3.12.8 (dalla 3.9.17) nel livello Lambda e nel programma di installazione. • Aggiorna NodeJS alla versione 20.18.3 (dalla 18.20.3) nel livello Lambda e nel programma di installazione.	

Modifica	Descrizione	Data
	• Rimuovi la generazione di chiavi DSA per i nodi di accesso come DSA, che non è più supportata in OpenSSH 9.7+. • Imposta l'ID dell'istanza e le informazioni sul tipo di istanza in Slurm all'avvio dei nodi di calcolo. • Installa i driver NVIDIA senza l'opzione 'no-cc-version-check', che ora è obsoleta nel programma di installazione NVIDIA. • Aggiungi un validatore per applicare fino a 10 pool di nodi di accesso. • Aggiorna la dimensione predefinita del volume root a 45 GB. Correzioni di bug • Rimuovi l'uso di cfn-init per il bootstrap dei nodi di calcolo per ridurre i tempi di scalabilità dei nodi. • Risolve un problema che causa l'errore di avvio del nodo di calcolo quando viene utilizzato un proxy. • Su Ubuntu 22.04, installa il driver Nvidia con la stessa versione del compilatore	

Modifica	Descrizione	Data
	utilizzata per compilare il kernel per evitare errori di installazione.- Risolvi l'esecuzione dell'override del pacchetto aws-parallelcluster-node solo sul nodo principale durante l'aggiornamento. • Risolve un problema per cui i lavori containerizzati eseguiti in un ambiente multiutente (integrato con Active Directory) non funzionavano. Pyxis/Enroot • Risolve l'utilizzo di authselect che causa errori di bootstrap del nodo su Rocky 9.5+ quando viene utilizzato il servizio di directory.	

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.12.0	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Miglioramenti: • Aggiungi una nuova sezione di configurazione dell'immagine di build <code>Build/Installation</code> per attivare le installazioni del software on/off NVIDIA e del client Lustre. Per impostazione predefinita, il software NVIDIA, sebbene incluso nelle ParallelCluster AMI ufficiali, non viene installato da <code>build-image</code>. Per impostazione predefinita, il client Lustre è installato. • I comandi <code>export-cluster-logs</code> CLI ora <code>export-image-logs</code> possono esportare per impostazione predefinita i log nel bucket predefinito o nel ParallelCluster <code>Customs3bucket</code> se specificato nella configurazione. • Estendi il supporto di Amazon DCV a Ubuntu2204 su istanze ARM.	19 dicembre 2024

Modifica	Descrizione	Data
	Modifiche: • Aggiorna il driver NVIDIA alla versione 550.127.08 (da 550.90.07). Questo risolve un problema noto di NVIDIA. Per ulteriori informazioni, consulta Problemi noti nella documentazione di NVIDIA Data Center. • Aggiorna Amazon DCV alla versione 2024.0-18131-1. • server: 2024.0-18131-1 • dcv: 2024.0.631-1 • gl: 2024.0.1078-1 • visualizzatore web: 2024.0-18131-1 • Aggiorna il programma di installazione EFA a. 1.36.0 • Efa-driver: efa-2.13.0-1 • Efa-config: efa-config-1.17-1 • Efa-profile: efa-profile-1.7-1 • Libfabric-aws: libfabric-aws-1.22.0-1 • Rdma-core: rdma-core-54.0-1	

Modifica	Descrizione	Data
	• Open MPI: e openmpi40-aws-4.1.7-1 openmpi50-aws-5.0.5 • Auto-restart slurmctld in caso di errore. • Aggiorna mysql-community-client alla versione 8.0.39. • Rimuovi il supporto per Python 3.7 e 3.8, che sono fuori uso. Correzioni di bug • Risolve un problema per cui le modifiche nella sequenza degli script di azioni personalizzate non venivano rilevate durante gli aggiornamenti del cluster. • Aggiungi le autorizzazioni mancanti per l' AWS ParallelCluster API per creare i ruoli collegati al servizio per Elastic Load Balancing e Auto Scaling, necessari per distribuire i nodi di accesso. • Risolve un problema relativo al modo in cui otteniamo la regione durante la gestione dei volumi in modo che possa gestire correttamente la zona locale.	

Modifica	Descrizione	Data
	• Risolve un problema per cui l'aggiunta di file system EFS con <code>AccessPointIds</code> durante un aggiornamento non andava a buon fine. • Risolve un problema per cui, quando si utilizza PCAPI, l'aggiornamento del cluster poteva fallire durante l'aggiornamento di un parametro che non è di tipo <code>String</code> (ad esempio,). <code>MaxCount</code> • Quando si monta un OpenZFS esterno, non è più necessario impostare le regole in uscita per le porte 111, 2049, 20001, 20002, 20003.	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.11.1	Funzionalità: - Pyxis è ora disabilitato per impostazione predefinita, quindi deve essere abilitato manualmente come documentato nella documentazione del prodotto. - Aggiorna il runtime di Python alla versione 3.12 in ParallelCluster Lambda Layer. - Rimuovi il pinning della versione di setuptools alla versione precedente alla 70.0.0. - Aggiorna libjwt alla versione 1.17.0. Changelog completo Correzioni di bug - Risolve un problema nel modo in cui configuriamo il plugin Pyxis Slurm ParallelCluster che può causare errori nell'invio dei lavori. - Risolve un problema che causava la mancata implementazione nelle configurazioni con nodi di accesso aggiungendo le autorizzazioni mancanti	21 ottobre 2024

Modifica	Descrizione	Data
	richieste dai nodi di accesso nel modello pubblico di politiche. https://github.com/aws/aws-parallelcluster/issues/6483	

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.11.0	Miglioramenti • Aggiunge il supporto per azioni personalizzate sui nodi di accesso. • Consenti la connessione DCV ai nodi di accesso. • Aggiunge il supporto per la regione ap-southeast-3. • Aggiungi gruppi di sicurezza al sistema di bilanciamento del carico di rete del nodo di accesso. • Aggiungi la AllowedIps configurazione per i nodi di accesso. • Aggiungi una nuova configurazione SharedStorage/EfsSettings/AccessPointId per specificare un punto di accesso EFS opzionale per un montaggio • Consenti fino a 10 pool di nodi di accesso. • Installa enroot e pyxis nelle AMI ufficiali di pcluster Modifiche • [INTERRUZIONE] Il loginNodes campo restituito dall'API	26 settembre 2024

Modifica	Descrizione	Data
	<code>DescribeCluster</code> e dal comando CLI <code>describe-cluster</code> è stato modificato da dizionario a array per supportare più pool di nodi di accesso. Questa modifica interrompe la compatibilità con le versioni precedenti, rendendo queste operazioni incompatibili con i cluster distribuiti con versioni precedenti. • Aggiorna Slurm alla versione 23.11.10 (dalla 23.11.7). • Aggiorna Pmix a 5.0.3 (da 5.0.2). • Aggiorna il programma di installazione EFA a. 1.34.0 • Efa-driver: efa-2.10.0-1 • Efa-config: efa-config-1.17-1 • Efa-profile: efa-profile-1.7-1 • Libfabric-aws: libfabric-aws-1.22.0-1 • Rdma-core: rdma-core-52.0-1 • Open MPI: e openmpi40-aws-4.1.6-3	

Modifica	Descrizione	Data
	<code>openmpi50-aws-5.0.3-11</code> • Aggiorna il driver NVIDIA alla versione 550.90.07 (da 535.183.01). • Aggiorna CUDA Toolkit alla versione 12.4.1 (dalla 12.2.2). • Aggiorna Python alla versione 3.9.20 (dalla 3.9.19). • Aggiorna la libreria Intel MPI alla versione 2021.13.1.769 (dalla 2021.12.1.8). Correzioni di bug • Correggi il validatore <code>EfaPlacementGroupValidator</code> in modo che non suggerisca di configurare un gruppo di posizionamento quando vengono utilizzati i blocchi di capacità. • Risolvete gli errori occasionali di creazione dei cluster assicurando che i file system FSx for Lustre vengano creati secondo le regole del gruppo di sicurezza. • Risolve l'errore di eliminazione del cluster quando il	

Modifica	Descrizione	Data
	gruppo di posizionamento è abilitato. • Risolve il problema dei nodi di accesso contrassegnati come non integri quando si limitava l'accesso SSH. • Correggi <code>retrieve_supported_regions</code> in modo che possa ottenere l'URL S3 corretto. • Correzione dell'uso <code>describe_images</code> dell'impaginazione. • Risolve il <code>No route tables found</code> bug che si verificava quando si specificava la sottorete VPC predefinita su. <code>LoginNodes/Networking/SubnetIds</code>	
AWS ParallelCluster è stata rilasciata la versione 3.10.1	Correzioni di bug • Risolve l'errore di creazione dell'immagine nelle regioni della Cina.	8 luglio 2024

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.10.0	Miglioramenti: • Aggiungi una nuova sezione di configurazione <code>Scheduling/SlurmSettings/ExternalSlurmdbd</code> per connettere il cluster a un <code>Slurmdbd</code> esterno. • Aggiungi il supporto per <code>build-image</code> nelle regioni AWS Top Secret e Secret. AWS • Aggiungi il supporto per Amazon Linux 2023. • Aggiungi il supporto per <code>price-capacity-optimized</code> as an <code>AllocationStrategy</code> . • Aggiungi un validatore per impedire l'uso di gruppi di posizionamento con blocchi di capacità. Modifiche: • CentOS 7 non è più supportato. • Aggiorna Cinc Client alla versione 18.4.12 dalla 18.2.7. • Aggiorna munge alla versione 0.5.16 (dalla 0.5.15).	27 giugno 2024

Modifica	Descrizione	Data
	• Aggiorna Pmix a 5.0.2 (da 4.2.9). • Aggiorna le dipendenze dei libri di cucina di terze parti: • apt-7.5.22 (da apt-7.5.14) • openssh-2.11.12 (da openssh-2.11.3) • Rimuovi il ricettario di terze parti: selinux-6.1.12. • Aggiorna il programma 1.32.0 di installazione EFA a. • Efa-driver: efa-2.8.0-1 • Efa-config: efa-config-1.16-1 • Efa-profile: efa-profile-1.7-1 • Libfabric-aws: libfabric-aws-1.21.0-1 • Rdma-core: rdma-core-50.0-1 • Open MPI: e openmpi40-aws-4.1.6-3 openmpi50-aws-5.0.2-12 • Aggiorna il driver NVIDIA alla versione 535.183.01 (da 535.154.05).	

Modifica	Descrizione	Data
	• Aggiorna Python alla versione 3.9.19 (dalla versione 3.9.17). • Aggiorna la libreria Intel MPI alla versione 2021.12.1.8 (da 2021.9.0.43482). Correzioni di bug • Correggi la configurazione di Data Repository Associations da creare e facoltativa. <code>AutoExportPolicy</code> <code>AutoImportPolicy</code> • È stato risolto un problema durante l'eliminazione del cluster che ora completa la pulizia del parco di elaborazione quando le istanze sono chiuse o terminate. Questo serve a evitare errori di eliminazione dei cluster per i tipi di istanze con cicli di terminazione più lunghi. • Consenti l'abilitazione della dashboard di cloudwatch e la disabilitazione degli allarmi nella Monitoring sezione della configurazione del cluster. • Consenti a ParallelCluster Custom Resource di sopprimere i validatori utilizzando <code>PclusterC</code>	

Modifica	Descrizione	Data
	luster/SuppressValidators • Rimozione <code>/etc/profile.d/pcluster.sh</code> in modo che non venga eseguita a ogni accesso utente e non <code>cfn_bootstrap_virtualenv</code> venga aggiunta nella variabile di ambiente <code>PATH</code>. • Correggi le specifiche dell' ParallelCluster API sostituendo il campo <code>failureReason</code> con <code>failures</code> in <code>DescribeCluster</code> risposta. • Correggi le specifiche dell' ParallelCluster API aggiungendo lo stato dello CloudFormation stack mancante: <code>IMPORT_*</code>, e. <code>REVIEW_IN_PROGRESS</code> <code>UPDATE_FAILED</code> • Risolve un problema che impediva agli aggiornamenti del cluster di includere i file system EFS con crittografia in transito. • Risolve un problema che impediva il riavvio dei servizi <code>slurmctld</code> e <code>slurmd</code> al riavvio del nodo principale quando EFS viene utilizzato per dati interni condivisi.	

Modifica	Descrizione	Data
	- Sui sistemi Ubuntu, rimuovi la configurazione logrotate predefinita per i file di registro cloud-init che si sono scontrati con la configurazione proveniente da ParallelCluster - Risolve l'errore di creazione dell'immagine con RHEL 8.10 o versione successiva.	
AWS ParallelCluster è stata rilasciata la versione 3.9.3	Per eseguire l'aggiornamento, digitare <code>sudo pip install --upgrade aws-parallelcluster</code> Funzionalità: - È stato aggiunto il supporto FSx Lustre come tipo di archiviazione condivisa <code>inus-iso-east-1</code>. Correzioni di bug - Rimuovi <code>cloud_dns</code> dalla <code>SlurmctlParameters</code> configurazione di Slurm per evitare problemi di fanout di Slurm. Questo non è necessari o, poiché impostiamo gli indirizzi IP all'avvio dell'istanza.	19 giugno 2024

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.9.2	Funzionalità: • Aggiorna Slurm alla versione 23.11.7 (dalla 23.11.4). • Per maggiori dettagli, vedi on. CHANGELOG 3.9.2 GitHub	28 maggio 2024
AWS ParallelCluster è stata rilasciata la versione 3.9.1	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Correzioni di bug • Rimuove l'eliminazione ricorsiva della cartella di archiviazione condivisa mountdir quando si smontano i filesystem come parte dell'operazione update-cluster.	11 aprile 2024

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.9.0	Per eseguire l'aggiornamento, inserisci quanto segue: <code>sudo pip install --upgrade aws-parallelcluster</code> Miglioramenti: • Aggiungi il parametro di configurazione <code>DeploymentSettings/DefaultUserHome</code> per consentire agli utenti di spostare la home directory dell'utente predefinito <code>/local/home</code> in <code>/home</code> (impostazione predefinita). • Autorizza <code>MinCount</code> l'<code>MaxCount</code> aggiornamento <code>Queue</code> e la <code>ComputeResource</code> configurazione dei parametri senza la necessità di interrompere il parco di elaborazione. Ora è possibile aggiornarli impostando su <code>Scheduling/SlurmSettings/QueueUpdateStrategy</code> <code>TERMINATE</code>. AWS ParallelCluster terminerà solo i nodi rimossi durante un ridimensionamento della capacità del cluster eseguito tramite un aggiornamento del cluster.	5 marzo 2024

Modifica	Descrizione	Data
	• Consente di aggiornare lo storage condiviso esterno di tipo Efs, FsxLustre FsxOntap, FsxOpenZfs e FileCache senza sostituire la flotta di elaborazione e accesso. • Aggiungi il supporto per RHEL9. • Aggiungi il supporto per Rocky Linux 9 così come CustomAmi creato tramite il processo. build-image. Al momento non è disponibile alcuna AMI Linux AWS ParallelCluster Rocky9 ufficiale pubblica. • Rimuovi CommunicationParameters dall'elenco degli utenti negati delle impostazioni personalizzate di Slurm. • Aggiungi un DeploymentSettings/DisableSudoAccessForDefaultUser parametro per disabilitare l'accesso sudo dell'utente predefinito nei sistemi operativi supportati. • Modifiche ai file system FSx for Lustre ParallelCluster create da: Modificare la versione del server Lustre in 2.15.	

Modifica	Descrizione	Data
	• Aggiungi la possibilità di scegliere tra driver Nvidia Open e Closed Source durante la creazione di un'AMI, tramite l'attributo <code>['cluster']['nvidia']['kernel_open']</code> cookbook node. • * Aggiungi un'opzione di configurazione <code>clustermgtd ec2_instance_missing_max_count</code> per consentire un numero configurabile di tentativi per eventuali tentativi di Amazon EC2. Descrivi la coerenza delle istanze con le istanze di esecuzione. Modifiche • Aggiorna Slurm alla versione 23.11.4 (dalla versione 23.02.7). • Aggiorna il driver NVIDIA alla versione 535.154.05. • Aggiungi il supporto per Python 3.11, 3.12 in pcluster CLI e aws-parallelcluster-batch-cli. • Crea interfacce di rete utilizzando l'indice delle schede di rete dall'<code>NetworkCa</code>	

Modifica	Descrizione	Data
	rdIndex elenco delle risposte di Amazon DescribeInstances EC2, anziché eseguire il MaximumNetworkCards looping su un intervallo. - Non riesci a creare un cluster quando utilizzi i tipi di istanze P3, G3, P2 e G2 perché la loro architettura GPU non è compatibile con i driver Nvidia open source (OpenRM) introdotti nella versione 3.8.0. - Aggiorna le dipendenze dei libri di cucina di terze parti: nfs-5.1.2 (da nfs-5.0.0) - Aggiorna il programma di installazione EFA a 1.30.0. - Efa-driver: efa-2.6.0-1 - Efa-config: efa-config-1.15-1 - Efa-profile: efa-profile-1.6-1 - Libfabric-aws: libfabric-aws-1.19.0 - Rdma-core: rdma-core-46.0-1 - Open MPI: e openmpi40-aws-4.1.6-2	

Modifica	Descrizione	Data
	openmpi50-aws-5.0.0-11 • Aggiorna NICE DCV alla versione 2023.1-16388. • server: 2023.1.16388-1 • dcv: 2023.1.565-1 • gl: 2023.1.1047-1 • visualizzatore web: 2023.1.16388-1 Correzioni di bug • Risolve il problema che faceva fallire il processo quando veniva inviato come utente di Active Directory dai nodi di accesso. Il problema era causato da una configurazione incompleta dell'integrazione con l'Active Directory esterno sul nodo principale. • Rifattorizza le politiche IAM definite nel CloudFormation modello parallelclutser-policies.yaml per prevenire errori di implementazione delle API causati da policy che superano i limiti IAM. ParallelCluster • Risolve il problema che impediva l'avvio dei nodi	

Modifica	Descrizione	Data
	di accesso quando il nodo principale impiegava più tempo del previsto per scrivere le chiavi. Per i dettagli sulle modifiche , consulta CHANGELOG i file per il pacchetto aws-parallelcluster-ui su. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.8.0	AWS ParallelCluster rilasciata la versione 3.8.0. Miglioramenti: • Aggiungi il supporto per Amazon EC2 Capacity Blocks for ML. • Aggiungi il supporto per Rocky Linux 8 così come CustomAmi creato tramite build-image il processo. Al momento non è disponibile alcuna AMI Linux AWS ParallelCluster Rocky8 ufficiale pubblica. • Aggiungi un Scheduling/ScalingStrategy parametro per controllare la strategia di scalabilità del cluster da utilizzare all'avvio delle istanze Amazon EC2 per i nodi di calcolo Slurm. I valori possibili sono all-or-nothing ,greedy-all-or-nothing ,, e sono quelli predefiniti. best-effort all-or-nothing • Aggiungi un HeadNode/SharedStorageType parametro per utilizzare lo storage EFS anziché le esportazioni NFS dal	19 dicembre 2023

Modifica	Descrizione	Data
	volume root del nodo principale per le risorse del file system condivise all'interno del cluster: Intel ParallelCluster, Slurm e dati. /home Questo miglioramento riduce il carico sulla rete del nodo principale. • Consenti il montaggio /home come storage condiviso esterno EFS o FSx tramite la SharedStorage sezione del file di configurazione. • Aggiungi un nuovo parametro SlurmSettings/MungeKeySecretArn per consentire l'utilizzo di una chiave MUNGE esterna definita dall'utente da AWS Secrets Manager. • Aggiungi un Monitoring/Alarms/Enabled parametro per attivare Amazon CloudWatch Alarms per il cluster. • Aggiungi allarmi del nodo principale per monitorare e i controlli di integrità di Amazon EC2, l'utilizzo della CPU e lo stato generale del nodo principale e aggiungil	

Modifica	Descrizione	Data
	i alla dashboard creata con CloudWatch il cluster. • Aggiungi il supporto per le associazioni di repository di dati quando usi PERSISTENT_2 as DeploymentType per un FSx for Lustre gestito. • Aggiungi un Scheduling/SlurmSettings/Database/DatabaseName parametro per consentire agli utenti di specificare un nome personalizzato per il database sul server di database da utilizzare per la contabilità Slurm. • Crea InstanceType un parametro di configurazione opzionale durante la configurazione CapacityReservationTarget/CapacityReservationId nella risorsa di calcolo. • Aggiungi la possibilità di specificare un prefisso per i ruoli e le politiche IAM creati dall'API. AWS ParallelCluster • Aggiungi la possibilità di specificare un limite di autorizzazioni da applicare per i ruoli e le policy IAM	

Modifica	Descrizione	Data
	creati dall'API. AWS ParallelCluster Modifiche • Aggiorna Slurm a 23.02.7 (dal 23.02.6). • Aggiorna il driver NVIDIA alla versione 535.129.03. • Aggiorna CUDA Toolkit alla versione 12.2.2. • Usa i driver GPU NVIDIA open source (OpenRM) come modulo kernel NVIDIA per Linux invece del modulo closed source NVIDIA. • Rimuovi il supporto dei parametri di <code>all_or_nothing_batch</code> configurazione nel programma Slurm resume, a favore della nuova configurazione del cluster. <code>Scheduling/ScalingStrategy</code> • La convenzione di denominazione degli allarmi del cluster è stata modificata in <code>'[cluster-name] - [component-name] - [metric]'</code>. • Modifica i tipi di volume EBS predefiniti nelle regioni AWS Secret e Top Secret da gp2	

Modifica	Descrizione	Data
	a gp3, sia per i volumi root che per quelli aggiuntivi. - Il limite di autorizzazioni opzionale per l' AWS ParallelCluster API viene ora applicato a ogni ruolo IAM creato dall'infrastruttura API. - Aggiorna il programma di installazione EFA a. 1.29.1 - Efa-driver: efa-2.6.0-1 - Efa-config: efa-config-1.15-1 - Efa-profile: efa-profile-1.5-1 - Libfabric-aws: libfabric-aws-1.19.0-1 - Rdma-core: rdma-core-46.0-1 - Apri MPI: openmpi40-aws-4.1.6-1 - Aggiorna GDRCopy alla versione 2.4 in tutti i sistemi operativi supportati, ad eccezione di CentOS 7 in cui viene utilizzata la versione 2.3.1. - Aggiornamento alla versione 2.0-28. aws-cfn-bootstrap	

Modifica	Descrizione	Data
	• Aggiungi il supporto per Python 3.10 in aws-parallelcluster-batch-cli. Correzioni di bug • Risolve la configurazione di scalabilità incoerente dopo il rollback dell'aggiornamento del cluster quando si modifica l'elenco dei tipi di istanza dichiarati nelle risorse di calcolo. • Risolve la generazione di chiavi SSH degli utenti quando cambiano utente senza privilegi di root in cluster integrati con un server LDAP esterno tramite file di configurazione del cluster. • Risolvi la disattivazione della modalità di risparmio energetico Slurm durante l'impostazione. Scaledown IdleTime = -1 • Correggi il percorso codificato alla directory di installazione di Slurm nello script per Slurm Accounting. update_slurm_database_password.sh	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.7.2	AWS ParallelCluster rilasciata la versione 3.7.2. Modifiche: • Aggiorna Slurm alla versione 23.02.6.	25 ottobre 2023

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.7.1	AWS ParallelCluster rilasciata la versione 3.7.1. Modifiche: • Aggiorna Slurm a 23.02.5 (dal 23.02.4). • Aggiorna Pmix a 4.2.6 (da 3.2.3). • Aggiorna libjwt a 1.15.3 (da 1.12.0). • Aggiorna il programma di installazione EFA a, risolvendo il problema dei dati di scrittura RDMA in 1.26.1 P5. • Efa-driver: efa-2.5.0-1 . • Efa-config: efa-config-1.15-1 . • Efa-profile: efa-profile-1.5-1 . • Libfabric-aws: libfabric-aws-1.18.2-1 . • ERdma-core: rdma-core-46.0-1 . • Apri MPI: openmpi40-aws-4.1.5-4	22 settembre 2023

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.7.0	AWS ParallelCluster rilasciata la versione 3.7.0. Miglioramenti: • Supporta la configurazione delle priorità dei nodi statici e dinamici nelle risorse di calcolo utilizzando un AWS ParallelCluster file YAML di configurazione. • Aggiungi il supporto per Ubuntu 22. Le chiavi RSA non sono supportate per impostazione predefinita. • Aggiungi l'impostazione di configurazione della coda <code>JobExclusiveAllocation</code> per allocare i nodi in una partizione esclusivamente a un singolo processo in un dato momento. • Consenti il <code>aws-parallelcluster-node</code> pacchetto <code>Override</code> al momento della creazione del cluster e dell'aggiornamento del cluster. Per il nodo principale, questo vale per l'aggiornamento del cluster. Utile solo per scopi di sviluppo.	30 agosto 2023

Modifica	Descrizione	Data
	• Evita l'avvio del server NFS sui nodi di calcolo. • Aggiungi il supporto per i nodi di accesso. • Consenti la pianificazione basata sulla memoria quando vengono specificati più tipi di istanza per una risorsa di calcolo Slurm. • Aggiungi il supporto per montare Amazon File Cache esistente come storage condiviso. Modifiche: • Assegna ai nodi dinamici Slurm una priorità (peso) di 1000 per impostazione predefinita. In questo modo, Slurm può dare priorità ai nodi statici inattivi rispetto ai nodi dinamici inattivi. • Fai in modo che i <code>aws-parallelcluster-node</code> demoni gestiscano solo le partizioni Slurm gestite. AWS ParallelCluster • Aumenta l'intervallo di sondaggio di <code>EFS-utils watchdog</code> a 10 secondi. Questa modifica si applica quando <code>EncryptionInTransit</code> è impostata	

Modifica	Descrizione	Data
	<code>subtrue</code>, che è l'unica condizione che causa l'esecuzione del watchdog. • Aggiorna il programma di installazione EFA a. 1.25.1 • Efa-driver: efa-2.5.0-1 (da efa-2.1.1g) • Efa-config: efa-config-1.15-1 (da efa-config-1.13-1) • Efa-profile: efa-profile-1.5-1 (nessuna modifica) • Libfabric-aws: libfabric-aws-1.18.1-0 (da libfabric-aws-1.17.1-1) • Rdma-core: rdma-core-46.0-1 (da rdma-core-43.0-1) • Apri MPI: openmpi40-aws-4.1.5-4 (da openmpi40-aws-4.1.5-1) • Aggiorna Slurm alla versione 23.02.4. • Cambia il valore predefinito <code>Imds/ImdsSupport</code> da v1.0 a v2.0. • Deprecate Ubuntu 18. • Aggiorna la dimensione predefinita del volume root a	

Modifica	Descrizione	Data
	40 GB per tenere conto dei limiti su Centos 7. • Limita l'autorizzazione sul file tmp/wait_condition_handle /.txt all'interno del nodo principale in modo che solo root possa leggerlo. • Crea un file JSON di mappatura della lista di partizioni Slurm da utilizzar e dai demoni del pacchetto node per riconoscere le partizioni e le liste di nodi Slurm. PC-managed • Aggiorna il driver NVIDIA alla versione 535.54.03. • Aggiorna la libreria CUDA alla versione 12.2.0. • Aggiorna NVIDIA Fabric Manager a nvidia-fabricmanager-535. • Aggiorna ARM PL alla versione 23.04.1 solo per Ubuntu 22.04. • Aggiorna NICE DCV alla versione. 2023.0-15487 • Server: 2023.0.15487-1 • dcv: 2023.0.551-1 • gl: 2023.0.1039-1 • visualizzatore web: 2023.0.15487-1	

Modifica	Descrizione	Data
	Correzioni di bug • Aggiungi la convalida al <code>ScaledownIdleTime</code> valore, per evitare di impostare un valore inferiore a -1. • Risolvi l'errore di creazione del cluster con l'AMI Ubuntu Deep Learning su istanze GPU con DCV abilitato. • Risolve il problema che causava la creazione di politiche IAM sospese durante la creazione di un provider di risorse ParallelCluster CloudFormation personalizzato con <code>CustomLambdaRole</code> • Risolve un problema che causava il disallineamento del nome DNS dei nodi di calcolo su istanze con più interfacce di rete, quando si utilizzava uguale a <code>SlurmSettings/Dns/UseEc2Hostnames</code> <code>True</code> Per i dettagli sulle modifiche, consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-	

Modifica	Descrizione	Data
	cookbook e aws-parallelcluster-node su GitHub	
Rilascio solo della documentazione	AWS ParallelCluster Pubblicata la guida per l'utente specifica per la versione 3. Rilascio della sola documentazione: • AWS ParallelCluster la versione 3 ha una propria guida per l'utente separata.	17 luglio 2023

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.6.1	AWS ParallelCluster rilasciata la versione 3.6.1. Modifiche: • Evita la duplicazione dei nodi rilevati <code>clustermgtd</code> se i nodi di calcolo vengono aggiunti a più partizioni Slurm. Correzioni di bug • Rimuovi il codice rigido del nome del dispositivo del volume root (<code>/dev/sda1</code> e <code>/dev/xvda</code>) e recuperalo dalle AMI utilizzate durante <code>create-cluster</code> • Risolvi l'errore di creazione del cluster quando si utilizza una risorsa CloudFormation personalizzata con <code>ElasticIp</code> set to <code>True</code> • Risolvi gli errori di creazione e aggiornamento del cluster quando si utilizza una risorsa AWS CloudFormation personalizzata con file di configurazione di grandi dimensioni. • Risolve un problema che impediva la disattivazione	5 luglio 2023

Modifica	Descrizione	Data
	pt race della protezione su Ubuntu e che non consentiva il Cross Memory Attach (CMA) in libfabric. • Risolve rapidamente la logica di failover con capacità insufficiente quando si utilizzano più tipi di istanze e non viene restituita alcuna istanza. Per i dettagli sulle modifiche, consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.6.0	AWS ParallelCluster rilasciata la versione 3.6.0. Documentazione: • Aggiungi la documentazione per l'API della libreria AWS ParallelCluster Python. Miglioramenti: • Aggiungi il supporto per RHEL8. • Aggiungi una risorsa AWS CloudFormation personalizzata per la creazione e la gestione di cluster con CloudFormation • Aggiungi il supporto per la personalizzazione della configurazione del cluster Slurm nel file YAML di configurazione. AWS ParallelCluster • Crea Slurm con il supporto per LUA. • Aumenta il limite del numero massimo di code per cluster da 10 a 50. Ogni coda può contenere fino a 50 risorse di elaborazione. Ogni cluster può avere fino a 50 risorse di elaborazione.	22 maggio 2023

Modifica	Descrizione	Data
	• Aggiungi il supporto per specificare una sequenza di più script di azioni personalizzati per un evento configurato in <code>OnNodeStart</code>, <code>OnNodeConfigured</code>, e parametri. <code>OnNodeUpdated</code> • Aggiungi una nuova sezione di configurazione <code>HealthChecks /Gpu</code>, per applicare i controlli di integrità della GPU su un nodo di calcolo prima dell'esecuzione di un processo. • Aggiungi il supporto per Tags nella configurazione <code>SlurmQueues</code> e <code>SlurmQueues /ComputeResources</code>. • Aggiungi il supporto per DetailedMonitoring nella <code>Monitoring</code> configurazione. • Aggiungi <code>mem_used_percent</code>, <code>disk_used_percent</code> parametri per il monitoraggio della memoria del nodo principale e dell'utilizzo del disco del volume root nella AWS ParallelCluster CloudWatch dashboard e imposta allarmi	

Modifica	Descrizione	Data
	per il monitoraggio di queste metriche. • Aggiungi il supporto per la rotazione dei log gestiti. AWS ParallelCluster • Tieni traccia degli errori più comuni dei nodi di elaborazione e del periodo di inattività a più lungo dei nodi dinamici nella dashboard. CloudWatch • Imponi al server di autenticazione DCV di utilizzare almeno TLS-1.2 il protocollo durante la creazione del socket SSL. • Installa il pacchetto NVIDIA Data Center GPU Manager (DCGM) su tutti i sistemi operativi supportati tranne <code>aarch64 centos7</code> <code>alinux2</code> • Carica il modulo del kernel nvidia-uvm per impostazione predefinita per fornire la funzionalità Unified Virtual Memory (UVM) al driver CUDA. • Installa NVIDIA Persistence Daemon come servizio di sistema. Modifiche:	

Modifica	Descrizione	Data
	• Aggiorna Slurm alla versione 23.02.2 (dalla versione 22.05.8). • Aggiorna munge alla versione 0.5.15 (dalla versione 0.5.14). • Imposta lo Slurm <code>TreeWidth</code> su 30. • Imposta Slurm prolog e epilog le configurazioni nella directory di destinazione e rispettivamente. <code>/opt/slurm/etc/scripts/prolog.d/</code> <code>/opt/slurm/etc/scripts/epilog.d/</code> • Imposta Slurm su un massimo <code>BatchStartTimeout</code> di 3 minuti per l'esecuzione degli Prolog script durante la registrazione dei nodi di calcolo. • Aumenta il valore predefinito <code>RetentionInDays</code> dei CloudWatch log da 14 a 180 giorni. • Aggiorna il programma di installazione EFA a. 1.22.1 • DMS: 2.8.3-2 • Efa-driver: efa-2.1.1g (nessuna modifica)	

Modifica	Descrizione	Data
	- Efa-config: efa-config-1.13-1 (nessuna modifica) - Efa-profile: efa-profile-1.5-1 (nessuna modifica) - Libfabric-aws: libfabric-aws-1.17.1-1 (dal libfabric-aws-1.17.0-1) - Rdma-core: rdma-core-43.0-1 (nessuna modifica) - Apri MPI: openmpi40-aws-4.1.5-1 (nessuna modifica) - Aggiorna la versione del client Lustre 2.12 su Amazon Linux 2. Il client Lustre 2.12 è stato installato su Ubuntu 20.04, 18.04 e CentOS >= 7.7. - Aggiorna la versione del client Lustre a 2.10.8 CentOS 7.6. - Aggiorna il driver NVIDIA alla versione 470.182.03 (dalla versione 470.141.03). - Aggiorna NVIDIA Fabric Manager alla versione 470.182.03 (dalla versione 470.141.03).	

Modifica	Descrizione	Data
	• Aggiorna NVIDIA CUDA Toolkit alla versione 11.8.0 (dalla versione 11.7.1). • Aggiorna l'esempio di NVIDIA CUDA alla versione 11.8.0. • Aggiorna la libreria Intel MPI alla versione 2021 Update 9 (dalla versione 2021 Update 6). Per ulteriori informazioni, vedere Intel® MPI Library 2021 Update 9. • Aggiorna NICE DCV alla versione 2023.0-15022 (dalla versione 2022.2-14521). • server: 2023.0.15022-1 (dalla versione 2022.2-14521-1). • xdcv: 2023.0.547-1 (dalla versione 2022.2.519-1). • gl: 2023.0.1027-1 (dalla versione 2022.2.1012-1). • web_viewer: 2023.0.15022-1 (dalla versione 2022.2.14521-1).	

Modifica	Descrizione	Data
	• Aggiorna <code>aws-cfn-bootstrap</code> alla versione. <code>2.0-24</code> • Aggiorna l'immagine utilizzata dall' CodeBuild ambiente durante la creazione di immagini di container per i cluster AWS Batch: • <code>aws/codebuild/amazonlinux2-x86_64-standard:4.0</code> (da <code>aws/codebuild/amazonlinux2-x86_64-standard:3.0</code>). • <code>aws/codebuild/amazonlinux2-aarch64-standard:2.0</code> (da <code>aws/codebuild/amazonlinux2-aarch64-standard:1.0</code>). Correzioni di bug • Correggi i validatori dei gruppi di sicurezza di rete Amazon EFS e Amazon FSx per evitare di segnalare falsi errori. • Corregge i tag mancanti delle risorse create da Image Builder durante	

Modifica	Descrizione	Data
	build-image l'operazione. • Correggi la politica di aggiornamento MaxCount per eseguire sempre confronti numerici sulla proprietà. MaxCount • Correggi l'allineamento IP sulle istanze dei nodi di calcolo con più schede di rete. • Risolve la sostituzione di StoragePass in slurm_parallelcluster_slurmdbd.conf quando viene eseguito un aggiornamento dei parametri di coda e le configurazioni di contabilità Slurm non vengono aggiornate. • Risolve il problema che causa la creazione di gruppi di sicurezza sospesi durante la creazione di un cluster con un file system EFS esistente. • Risolve il problema che causava il fallimento cfn-hup del demone al riavvio. • Considerate i nodi dinamici con INVALID_REG flag come errori di bootstrap	

Modifica	Descrizione	Data
	per la modalità protetta Slurm. I nodi statici che non registrano la registrazione con Slurm vengono già considerati come errori di bootstrap dopo il. <code>node_replacement_timeout</code> Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.5.1	AWS ParallelCluster rilasciata la versione 3.5.1. Miglioramenti: • Aggiungi un eseguibile di installazione pcluster CLI autonomo. Modifiche: • Aggiorna il programma di installazione EFA a. 1.22.0 • Efa-driver: efa-2.1.1g (da efa-2.1.1-1) • Efa-config: efa-config-1.13-1 (da efa-config-1.12-1) • Efa-profile: (nessuna modifica) efa-profile-1.5-1 • Libfabric-aws: libfabric-aws-1.17.0-1 (da libfabric-aws-1.16.1amzn3.0-1) • Rdma-core: rdma-core-43.0-1 (nessuna modifica) • Apri MPI: openmpi40-aws-4.1.5-1 (da openmpi40-aws-4.1.4-3)	29 marzo 2023

Modifica	Descrizione	Data
	Aggiorna NICE DCV alla versione. 2022.2-14521 • server: 2022.2.14521-1 • dcv: 2022.2.519-1 • gl: 2022.2.1012-1 • visualizzatore web: 2022.2.14521-1 Correzioni di bug • Risolvi i potenziali errori di avvio dei nodi causati dalla corrispondenza dei pattern tra MountDir e /etc/exports durante la rimozione di volumi Amazon EBS condivisi come parte di un aggiornamento del cluster. • Correzione per impedire il troncamento dei file di compute_console_output registro a ogni iterazione. clustermgtd Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.5.0	AWS ParallelCluster rilasciata la versione 3.5.0. Miglioramenti: • Accedi e gestisci i cluster con l'AWS ParallelCluster interfaccia utente. • Aggiungi AWS ParallelCluster policy con versioni diverse in un CloudFormation modello a cui puoi fare riferimento nei tuoi carichi di lavoro. • Aggiungi una libreria AWS ParallelCluster Python che puoi usare con il tuo codice. • Aggiungi la registrazione dell'output della console del nodo di calcolo ad Amazon in caso di errore di CloudWatch avvio del nodo di calcolo. • Aggiungi il campo Failures contenente il codice di errore e il motivo da inviare quando la creazione del cluster non riesce <code>describe-cluster</code>. • Aggiungi validatori per impedire l'iniezione di stringhe dannose durante	20 febbraio 2023

Modifica	Descrizione	Data
	la chiamata al modulo del sottoprocesso. - Se lo stato del cluster cambia PROTECTED durante il provisioning di nodi statici, fallisce la creazione del cluster. Modifiche: - Aggiornamento alla versione Slurm 22.05.8 (dalla versione) 22.05.7 - Aggiorna il programma di installazione EFA a. 1.21.0 - Efa-driver: efa-2.1.1-1 (da) efa-2.1 - Efa-config: efa-config-1.12-1 (da efa-config-1.11-1) - Efa-profile: (nessuna modifica) efa-profile-1.5-1 - Libfabric-aws: libfabric-aws-1.16.1amzn3.0-1 (dalibfabric-aws-1.16.1) - Rdma-core: rdma-core-43.0-1 (dardma-core-43.0-2)	

Modifica	Descrizione	Data
	• Apri MPI: <code>openmpi40-aws-4.1.4-3</code> (nessuna modifica) • Rendi i log del controller Slurm più dettagliati e abilita la registrazione aggiuntiva per il plug-in Slurm power save. Correzioni di bug • Correggi la creazione del database cluster verificando che il nome del cluster non superi i 40 caratteri quando la contabilità Slurm è abilitata. • Risolve un problema <code>clustermgtd</code> che causava la sostituzione dei nodi di calcolo, riavviati tramite Slurm, in caso di esito negativo dei controlli dello stato delle istanze Amazon EC2. • Risolve un problema che impediva l'avvio dei nodi di elaborazione, con prenotazioni di capacità condivise da altri account, a causa di una politica IAM errata sul nodo principale.	

Modifica	Descrizione	Data
	Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook, aws-parallelcluster-node e aws-parallelcluster-ui su. GitHub	
AWS ParallelCluster è stata rilasciata la versione 3.4.1	AWS ParallelCluster rilasciata la versione 3.4.1. Correzioni di bug • Risolve un problema dello scheduler Slurm che poteva causare l'applicazione errata degli aggiornamenti al registro interno dei nodi di calcolo. Di conseguenza, in caso di questo problema, le istanze EC2 potrebbero o non essere disponibili o essere supportate da un tipo di istanza errato. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	13 gennaio 2023

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.4.0	AWS ParallelCluster rilasciata la versione 3.4.0. Miglioramenti: • Aggiungi il supporto per l'avvio di nodi su più zone di disponibilità per aumentare la disponibilità della capacità. • Aggiungi il supporto per specificare più sottoreti per ogni coda per aumentare la disponibilità della capacità. • Aggiungi un nuovo parametro di configurazione in Iam/ResourcePrefix per specificare un prefisso per il percorso e il nome delle risorse IAM create da AWS ParallelCluster • Aggiungi una nuova sezione di configurazione DeploymentSettings/LambdaFunctionsVpcConfig per specificare la configurazione Vpc utilizzata dalle funzioni AWS ParallelCluster Lambda. • Aggiungi la possibilità di specificare uno script personalizzato da eseguire nel nodo principale durante un aggiornamento del	22 dicembre 2022

Modifica	Descrizione	Data
	cluster. Lo script può essere specificato con HeadNode/CustomActions/OnNodeUpdated quando si utilizza Slurm come scheduler. Modifiche: • Rimuovi la creazione di destinazioni di montaggio Amazon EFS per i file system esistenti. • Monta i file system EFS utilizzando <code>amazon-efs-utils</code>. I file system EFS possono essere montati utilizzando la crittografia in transito e un utente autorizzato IAM. • Installa <code>stunnel 5.67</code> su CentOS7 e Ubuntu per supportare la crittografia EFS in transito. • Aggiorna il programma di installazione EFA a (da) <code>1.20.0 1.18.0</code> • Efa-driver: <code>efa-2.1 (da efa-1.16.0-1</code> • Efa-config: <code>efa-config-1.11-1</code> (nessuna modifica)	

Modifica	Descrizione	Data
	- Efa-profile: efa-profile-1.5-1 (nessuna modifica) - Libfabric-aws: libfabric-aws-1.16.1 (da libfabric-aws-1.16.0~amzn4.0-1) - Rdma-core: rdma-core-43.0-2 da (rdma-core-41.0-2) - Apri MPI: openmpi40-aws-4.1.4-3 da () openmpi40-aws-4.1.4-2 - Aggiorna Slurm alla versione 22.05.7 (da). 22.05.5 - Aggiorna Python a and. 3.9.16 3.7.16 (da 3.9.15 e 3.7.13). - Con Slurm 22.05.7, i nodi dinamici in IDLE +CLOUD+COMPLETING +POWER_DOWN+NOT _RESPONDING stato non sono considerati malsani. Per i dettagli delle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-	

Modifica	Descrizione	Data
	cookbook e aws-parallelcluster-node su GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.3.1	AWS ParallelCluster rilasciata la versione 3.3.1. Modifiche: • Le AMI ufficiali AWS ParallelCluster del prodotto sono ora disponibili dopo due anni di deprecazione di Amazon EC2. • Aumenta la dimension e della memoria dell' AWS ParallelCluster API Lambda a 2048 per ridurre le penalità di avvio a freddo ed evitare i timeout. Correzioni di bug • Impedisci la sostituzione dei file system FSx for Lustre gestiti e la perdita di dati sugli aggiornamenti del cluster che includono modifiche all'ID della sottorete della flotta di elaborazione. • SharedStorageDeletionPolicy si applica alle azioni di aggiornamento del cluster. Per i dettagli delle modifiche , consulta il CHANGELOG file	2 dicembre 2022

Modifica	Descrizione	Data
	per il pacchetto aws-parallelcluster su. GitHub	
AWS ParallelCluster documentazione (solo nota hpc6id)	AWS ParallelCluster aggiornamento solo della documentazione • AWS ParallelCluster non supporta il tipo di istanza hpc6id per l'impostazione /. HeadNodeInstanceType	2 dicembre 2022

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.1.5	AWS ParallelCluster rilasciata la versione 3.1.5. Miglioramenti: • Risolve il problema Slurm che impedisce la chiusura dei nodi inattivi. • Aggiorna il programma di installazione EFA alla versione 1.18.0 • Efa-driver: efa-1.16.0-1 • Efa-config: (da) efa-config-1.11-1 efa-config-1.9-1 • Efa-profile: efa-profile-1.5-1 (nessuna modifica) • Libfabric-aws: libfabric-aws-1.16.0~amzn4.0-1 (dalibfabric-1.13.2). • Rdma-core: rdma-core-41.0-2 (dardma-core-37.0) • Apri MPI: openmpi40-aws-4.1.4-2 (daopenmpi40-aws-4.1.1-2) Modifiche:	16 novembre 2022

Modifica	Descrizione	Data
	• Aggiungi <code>lambda:ListTags</code> e <code>lambda:UntagResource</code> a quello <code>ParallelClusterUserRole</code> utilizzato dallo stack di AWS ParallelCluster API per un aggiornamento del cluster. • Aggiornare la libreria Intel MPI alla versione 2021 Update 6 (dalla versione 2021 Update 4). Per ulteriori informazioni, vedere Intel® MPI Library 2021 Update 6. • Aggiorna il driver NVIDIA alla versione 470.141.03 (da 470.103.01). • Aggiorna NVIDIA Fabric Manager alla versione 470.141.03 (da 470.103.01). Per i dettagli sulle modifiche, consulta i CHANGELOG file per i pacchetti <code>aws-parallelcluster</code>, <code>aws-parallelcluster-cookbook</code> e <code>aws-parallelcluster-node</code> su GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.3.0	AWS ParallelCluster rilasciata la versione 3.3.0. Miglioramenti: • Aggiungi il supporto per la configurazione di allocazione di più istanze per una risorsa di calcolo quando usi Slurm come scheduler . Per ulteriori informazioni, consulta Allocazione di più tipi di istanza con Slurm. • Aggiungi il supporto per l'aggiunta e la rimozione SharedStorage con un aggiornamento del cluster, utilizzando una configurazione aggiornata. Per ulteriori informazioni, consulta Archiviazione condivisa. • Aggiungi un nuovo parametro di configurazione <code>DeletionPolicy</code> per Efs e impostazioni di archiviazione FsxLustre condivisa per supportare la conservazione dello spazio di archiviazione. • Aggiunge il supporto per la contabilità Slurm con il nuovo parametro di configurazione Scheduling	2 novembre 2022

Modifica	Descrizione	Data
	g//SlurmSettingsDatabase. Per ulteriori informazioni, vedere Slurm accounting with. AWS ParallelCluster • Aggiungi il supporto per le prenotazioni On-Demand di capacità (ODCR) e i gruppi di risorse per la prenotazione della capacità. Per ulteriori informazioni, consulta Launch instances with On-Demand Capacity Reservations (ODCR). • Aggiungi un nuovo parametro di configurazione per specificare la versione IMDS da supportare in un cluster o crea un'infrastruttura di immagini nel cluster, Imds/, e build ImdsSupport, Imds/, configurazioni. ImdsSupport • Aggiungi il supporto per Networking/Placement Group nella sezione SlurmQueuesComputeResources • Aggiungi il supporto per le istanze con più interfacce di rete limitate a un solo ENI per dispositivo. • Migliora la convalida della rete per i file system Amazon EFS esterni	

Modifica	Descrizione	Data
	controllando il blocco CIDR nel gruppo di sicurezza collegato. • Aggiungi un validatore per verificare se i tipi di istanza configurati supportano i gruppi di posizionamento. • Configura i thread NFS in modo che siano min (256, max (8, num_cores * 4)) per garantire stabilità e prestazioni migliori. • Sposta l'installazione di NFS in fase di compilazione per ridurre i tempi di configurazione. • Abilita la crittografia lato server per l'argomento EcrlImageBuilder SNS che viene creato durante la distribuzione dell' AWS ParallelCluster API e viene utilizzato per notificare gli eventi di creazione dell'immagine docker. Modifiche: • Modifica il comportamento di SlurmQueues/Networking/PlacementGroup/Enabled. Ora crea un gruppo di posizionamento gestito unico per ogni	

Modifica	Descrizione	Data
	risorsa di calcolo anziché un singolo gruppo di posizionamento gestito per tutte le risorse di elaborazione. • Aggiunge il supporto per SlurmQueues/Networking/PlacementGroup/Name come metodo di denominazione preferito. • Sposta i tag del nodo principale da Launch Template alla definizione dell'istanza per evitare la sostituzione del nodo principale durante gli aggiornamenti dei tag. • Disabilita il multithreading tramite script eseguito <code>cloud-init</code> e non tramite <code>CpuOptions</code> set nel Launch Template. • Aggiorna Python alla versione 3.9 e NodeJS alla versione 16 nell'infrastruttura API, nel contenitore API Docker e nelle risorse Lambda del cluster. • Rimuovi il supporto per Python 3.6 in <code>aws-parallelcluster-batch-cli</code> • Aggiorna Slurm alla versione (da)22.05.5.21.08.8-2	

Modifica	Descrizione	Data
	• Aggiorna il driver NVIDIA alla versione 470.141.03 (da). 470.129.06 • Aggiorna NVIDIA Fabric Manager alla versione 470.141.03 (da 470.129.06). • Aggiorna NVIDIA CUDA Toolkit alla versione 11.7.1 (). from 11.4.4 • Aggiorna Python usato in AWS ParallelCluster virtualenvs da a. 3.7.13 3.9.15 • Aggiorna il programma di installazione EFA alla versione 1.18.0. • Efa-driver: efa-1.16.0-1 (nessuna modifica) • Efa-config: efa-config-1.11-1 (from efa-config-1.10-1) • Efa-profile: efa-profile-1.5-1 (nessuna modifica) • Libfabric-aws: libfabric-aws-1.16.0~amzn4.0-1 (da libfabric-aws-1.16.0~amzn2.0-1).	

Modifica	Descrizione	Data
	- Rdma-core: rdma-core-41.0-2 (da rdma-core-37.0) - Apri MPI: openmpi40-aws-4.1.4-2 (da openmpi40-aws-4.1.1-2) - Aggiorna NICE DCV alla versione 2022.1-13300 (da 2022.0-12760). - Abilita la soppressione del modulo. SingleSubnetValidator Queues - Non sostituite DRAIN i nodi quando i nodi sono attivi, poiché COMPLETING Epilog potrebbe essere ancora in esecuzione. Correzioni di bug - Risolve la convalida del parametro dei filtri nel AWS ParallelCluster ListClusterLogStreams comando in modo che fallisse quando vengono passati filtri errati. - Corregge la convalida del parametro SharedStorage/EfsSettings in modo che non riesca la convalida quando FileSyste	

Modifica	Descrizione	Data
	mId viene specificato insieme ad altri SharedStorageparametri/. EfsSettings In precedenza, FileSystem mId non era incluso. • Corregge l'aggiornamento del cluster quando si modifica l'ordine SharedStorageinsieme ad altre modifiche nella configurazione. • Correzione UpdateParallelClusterLambda Role nell' AWS ParallelCluster API in cui caricare i log. CloudWatch • Risolto il problema che Cinc non utilizzava il pacchetto di certificati CA locali durante l'installazione dei pacchetti prima dell'esecuzione di qualsiasi libro di cucina. • Risolve un problema nell'aggiornamento di Ubuntu con when is set. pcluster build-image Build:UpdateOsPackages:Enabled:true • Risolve l'analisi della configurazione del cluster YAML fallendo sulle chiavi duplicate.	

Modifica	Descrizione	Data
	Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	
AWS ParallelCluster alla documentazione è stato aggiunto solo il riferimento all'API.	AWS ParallelCluster aggiornamento solo della documentazione • È stato aggiunto il riferimento all'AWS ParallelCluster API della versione 3 alla documentazione.	27 ottobre 2022

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.2.1	AWS ParallelCluster rilasciata la versione 3.2.1. Miglioramenti: • Migliora la logica per associare le tabelle di routing dell'host alle diverse schede di rete per supportar e meglio le istanze Amazon EC2 con diverse NIC. Modifiche: • Aggiorna il driver NVIDIA alla versione 470.141.03. • Aggiorna NVIDIA Fabric Manager alla versione 470.141.03. • Disattiva le attività cron lavorative man-db e ciò mlocate potrebbe avere un impatto negativo sulle prestazioni dei nodi. • Aggiornare la libreria Intel MPI alla versione 2021.6.0.602. • Aggiorna Python da 3.7.10 a 3.7.13 in risposta a questo rischio per la sicurezza. Correzioni di bug	3 ottobre 2022

Modifica	Descrizione	Data
	• Evita di fallire <code>DescribeCluster</code> quando la configurazione del cluster non è disponibile. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti <code>aws-parallelcluster</code>, <code>aws-parallelcluster-cookbook</code> e <code>aws-parallelcluster-nodesu</code>. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.2.0	AWS ParallelCluster rilasciata la versione 3.2.0. Miglioramenti: • Aggiungi il supporto per la pianificazione basata sulla memoria in Slurm. • Configura la memoria reale dei nodi di calcolo nella configurazione del cluster Slurm. • Aggiungi un nuovo parametro di configurazione Scheduling/SlurmSettings/EnableMemoryBasedSchedulingperabilitare la pianificazione basata sulla memoria in Slurm. • Aggiungi un nuovo parametro di configurazione Scheduling/SlurmQueuesComputeResources/ScheduleableMemory per sovrascrivere il valore predefinito della memoria visualizzato dallo scheduler sui nodi di calcolo. • Migliora la flessibilità sugli aggiornamenti della	27 luglio 2022

Modifica	Descrizione	Data
	configurazione del cluster per evitare l'arresto e l'avvio dell'intero cluster quando possibile. Aggiungi il nuovo parametro di configurazione Scheduling SlurmSettings//QueueUpdateStrategy per impostare la strategia preferita da utilizzare quando i nodi di calcolo richiedono un aggiornamento e una sostituzione della configurazione. • Migliora il meccanismo di fail-over sulle risorse di elaborazione disponibili in caso di problemi di capacità insufficiente con le istanze Amazon EC2. Disattiva i nodi di calcolo entro un periodo di tempo configurabile quando l'avvio di un nodo fallisce a causa di una capacità insufficiente. • Aggiungi il supporto per montare i file system FSx for ONTAP e FSx for OpenZFS esistenti. • Aggiungi il supporto per montare più istanze di file system Amazon Elastic File System esistenti, FSx for Lustre, FSx for ONTAP e FSx for OpenZFS.	

Modifica	Descrizione	Data
	• Aggiungi il supporto per il tipo di distribuzione FSx for Lustre Persistent_2 durante la creazione di un nuovo file system. • Richiedi all'utente di abilitare EFA per i tipi di istanza supportati quando usa la procedura guidata di configurazione di pcluster. • Aggiungi il supporto per il riavvio dei nodi di calcolo utilizzando Slurm. • Migliora la gestione degli stati di alimentazione di Slurm per tenere conto anche dello spegnimento manuale dei nodi. • Installa NVIDIA GDRCopy 2.3 nelle AMI del prodotto per abilitare la copia della memoria GPU a bassa latenza. Modifiche: • Aggiorna il programma di installazione EFA alla versione 1.17.2. • Driver EFA: efa-1.16.0-1 • Configurazione EFA: efa-config-1.10-1	

Modifica	Descrizione	Data
	• Profilo EFA: efa-profile-1.5-1 • Libfabric: libfabric-aws-1.16.0~amzn2.0-1 • Codice RDMA: rdma-core-41.0-2 • Apri MPI: openmpi40-aws-4.1.4-2 • Aggiorna NICE DCV alla versione 2022.0-12760. • Aggiorna il driver NVIDIA alla versione 470.129.06. • Aggiorna NVIDIA Fabric Manager alla versione 470.129.06. • Cambia i tipi di volume EBS predefiniti da gp2 a gp3 sia nel volume root che in quello aggiuntivo. • Modifiche ai file system FSx for Lustre create da: AWS ParallelCluster • Modificare il tipo di distribuzione predefinito in. Scratch_2 • Cambia la versione del server Lustre in. 2.12 • Non richiede che Placement Group/Enabled sia impostato su true quando	

Modifica	Descrizione	Data
	si passa unPlacement Group /Idesistente. - Non consente l'impostazionePlacement Group /IdquandoPlacement Group /Enabledè impostato esplicitamente su. false - Aggiungi parallelcluster:cluster-name tag a tutte le risorse create da AWS ParallelCluster. - Aggiungi lambda:ListTags e ParallelClusterUserRole utilizza lambda:UntagResource lo stack AWS ParallelCluster API per l'aggiornamento del cluster. - Limita l'accesso IPv6 solo agli utenti root e amministratori del cluster, quando il parametro di configurazioneHeadNode/Imds/Secureabilitato. IMDS - Con un'AMI personalizzata, utilizza la dimensione del volume root AMI anziché quella ParallelCluster predefinita di 35 GiB. Il valore può essere modificato nel file di configurazione del cluster.	

Modifica	Descrizione	Data
	- Disattivazione automatica della flotta di elaborazione quando il parametro di configurazione <code>Scheduling //SlurmQueues ComputeResources /SpotPrice</code> è inferiore al prezzo minimo richiesto per l'evasione delle richieste Spot. - Visualizzazione <code>requested_value</code> e <code>current_value</code> valori nel set di modifiche quando si aggiunge o si rimuove una sezione durante un aggiornamento. - Disattiva <code>aws-ubuntu-eni-helper</code> il servizio, disponibile nelle AMI di Deep Learning, per evitare conflitti <code>configure_nw_interface.sh</code> durante la configurazione di istanze con più schede di rete. - Rimuovi il supporto per Python 3.6. - Imposta MTU su 9001 per tutte le interfacce di rete quando configuri istanze con più schede di rete.	

Modifica	Descrizione	Data
	• Rimuovi il punto finale quando configuri l'FQDN del nodo di calcolo. • Gestisci i nodi statici in <code>POWERING_DOWN</code> • Non sostituisce il nodo dinamico in <code>POWER_DOWN</code> quanto i lavori potrebbero essere ancora in esecuzione. • Riavvio <code>clustermgtd</code> e <code>slurmctld</code> daemon al momento dell'aggiornamento del cluster solo quando <code>Scheduling</code> i parametri vengono aggiornati nella configurazione del cluster. • File di aggiornamento <code>slurmctld</code> e <code>slurmd</code> <code>systemd</code> assistenza. • Limita l'accesso IPv6 a IMDS solo agli utenti amministratori root e del cluster, quando il parametro di configurazione <code>HeadNode/Imds/Secure</code> abilitato. • Imposta la configurazione <code>Slurm AuthInfo=cred_expire=70</code> per ridurre il tempo richiesto, i lavori devono attendere	

Modifica	Descrizione	Data
	prima di ricominciare quando i nodi non sono disponibili. • Aggiorna le dipendenze dei libri di cucina di terze parti: • apt-7.4.2 (da apt-7.4.0) • line-4.5.2 (dalla riga 4.0.1) • openssh-2.10.3 (da openssh-2.9.1) • pyenv-3.5.1 (da pyenv-3.4.2) • selinux-6.0.4 (da selinux-3.1.1) • yum-7.4.0 (da yum-6.1.1) • yum-epel-4.5.0 (da yum-epel-4.1.2) Correzioni di bug • Correggi il comportamento predefinito per saltare i passaggi di AWS ParallelCluster convalida e test quando crei un'AMI personalizzata. • Risolve la perdita di dati sulla gestione dei file. <code>computemgtd</code> • Risolve una condizione di gara che causava sporadicamente la chiusura immediata delle istanze avviate perché non erano	

Modifica	Descrizione	Data
	ancora disponibili nella risposta EC2. Describelnstances • Corregge il supporto per il DisableSimultaneousMultithreading parametro sui tipi di istanza con processori Arm. • AWS ParallelCluster Risolve l'errore di aggiornamento dello stack API durante l'aggiornamento da una versione precedente. Aggiungi il modello di risorse utilizzato per l'ListImagePipelineImages azione in. EcrImageDeletionLambdaRole • Correggi l' AWS ParallelCluster API aggiungendo le autorizzazioni mancanti necessarie per importare o esportare da Amazon S3 durante la creazione di un file system FSx for Lustre. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster aggiornamenti solo sulla documentazione da quest'anno ad oggi	AWS ParallelCluster aggiornamenti solo per la documentazione. Nuove sezioni: • Migliori pratiche: avvisi sul budget V3 • Migliori pratiche: spostare un cluster a una nuova versione AWS ParallelCluster secondaria o patch V3 • Lavorare con Amazon S3 V3 • Lavorare con Spot Instances V3 • modalità protetta del cluster Slurm V3 • AWS ParallelCluster risorse e etichettatura V3 • CloudWatch Pannello di controllo Amazon V3 • Integrazione con Amazon CloudWatch Logs V3 • Adattatore Elastic Fabric V3 • AWS ParallelCluster Personalizzazione AMI V3 • Avvia istanze con On-Demand Capacity Reservations (ODCR) V3 • Patch AMI e sostituzione delle istanze Amazon EC2 V3	6 luglio 2022

Modifica	Descrizione	Data
	• Come funziona la V3 AWS ParallelCluster • Configurazione della crittografia dello storage condiviso con una chiave AWS KMS V3 • Esecuzione di processi in un cluster con modalità di coda multipla V3 • Utilizzo dell'API V3 AWS ParallelCluster Aggiornamenti della sezione: • Migliori pratiche: prestazioni di rete V3: sono state aggiunte le migliori pratiche per l'utilizzo di Elastic Fabric Adaptor. • AWS Autorizzazioni di Identity and Access Management in AWS ParallelCluster V3: vari aggiornamenti e aggiunta di una politica utente AWS ParallelCluster pcluster aggiuntiva quando si utilizza Amazon FSx for Lustre. • AWS ParallelCluster risoluzione dei problemi V3: vari aggiornamenti.	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.1.4	AWS ParallelCluster rilasciata la versione 3.1.4. Miglioramenti: • Aggiungi la convalida per DirectoryService/PasswordSecretArn in modo che fallisca se il segreto non esiste. Aggiungi il supporto per abilitare l'autenticazione JWT Slurm. Modifiche: • Aggiorna Slurm alla versione 21.08.8-2. • Crea Slurm con il supporto JWT. • Non richiede PlacementGroup/Enabled per essere impostato su true quando si passa un/esistente. PlacementGroup Id • Aggiungi <code>lambda:TagResource</code> a <code>ParallelClusterUserRole</code> Used by ParallelCluster API stack per la creazione di cluster e la creazione di immagini. Correzioni di bug	16 maggio 2022

Modifica	Descrizione	Data
	• Risolvi la possibilità di esportare i log di un cluster quando si utilizza il <code>export-cluster-logs</code> comando con l'<code>--filters</code> opzione. • Correggi il punto di ingresso di AWS Batch Docker per utilizzare la directory <code>/home</code> condivisa per coordinare l'esecuzione del Multi-node-Parallel lavoro. • Reimposta l'indirizzo del nodo quando imposti il nodo statico non sano Slurm su Disattivo per evitare che il nodo statico non funzionasse con capacità insufficiente come nodo di errore di bootstrap. Per i dettagli sulle modifiche, consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.1.3	AWS ParallelCluster rilasciata la versione 3.1.3. Miglioramenti: • Esegui la creazione della chiave SSH insieme alla creazione della directory HOME, ad esempio durante l'accesso SSH, quando passi a un altro utente e quando esegui un comando come altro utente. • Aggiungi il supporto per FQDN e LDAP Distingui shed Names nel parametro di configurazione/. Directory ServiceDomainName Il nuovo validatore ora controlla entrambe le sintassi. • Il nuovo <code>update_directory_service_password.sh</code> script distribuito sul nodo principal e supporta l'aggiornamento manuale della password di Active Directory nella configurazione SSSD. La password viene recuperata dal AWS Secrets Manager come dalla configurazione del cluster.	20 aprile 2022

Modifica	Descrizione	Data
	• Aggiungi il supporto per implementare l'infrastruttura API in ambienti senza un VPC predefinito. Modifiche: • Disabilita C-States deep nelle AMI ufficiali x86_64 e nelle AMI create tramite <code>build-image</code> comando, per garantire alte prestazioni e bassa latenza. • Aggiornamenti dei pacchetti del sistema operativo e correzioni di sicurezza. • Modifica le immagini di base di Amazon Linux 2 per utilizzare le AMI con Kernel 5.10. Correzioni di bug • Correggi lo stack di immagini di compilazione <code>DELETE_FAILED</code> dopo che l'immagine è stata creata con successo, grazie alle nuove politiche di EC2 Image Builder. • Correggi il parametro di configurazione Directory Service/la DomainAdd rconversione nella proprietà	

Modifica	Descrizione	Data
	SSSD Idap_uri quando contiene più indirizzi di dominio. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster e aws-parallelcluster-cookbook su. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.1.2	AWS ParallelCluster rilasciata la versione 3.1.2. Modifiche: • Aggiorna Slurm alla versione 21.08.6 (da 21.08.5). Correzioni di bug • Correggi l'aggiornamento del <code>/etc/hosts</code> file sui nodi di calcolo quando un cluster viene distribuito in sottoreti senza accesso a Internet. • Correggi il bootstrap dei nodi di calcolo in modo che attenda l'inizializzazione delle unità temporanee prima di entrare nel cluster. Per i dettagli sulle modifiche, consulta i CHANGELOG file per il pacchetto aws-parallelcluster su GitHub	2 marzo 2022

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.1.1	AWS ParallelCluster rilasciata la versione 3.1.1. • Aggiungi il supporto per ambienti cluster di più utenti mediante l'integrazione con i domini Active Directory (AD) gestiti tramite AWS Directory Service. • Aggiungi il supporto per UseEc2Hostnames nel file di configurazione del cluster. Se impostato su true, usa i nomi host predefiniti di Amazon EC2 (ad esempio ip-1-2-3-4) per i nodi di calcolo. • Aggiungi il supporto per la creazione di cluster in sottoreti senza accesso a Internet. • Aggiungi il supporto per più tipi di istanze di calcolo per coda. • Aggiungi il supporto per la pianificazione delle GPU con Slurm su istanze ARM con schede NVIDIA. • Aggiungi flag abbreviati per <code>cluster-name</code> (<code>-n</code>), <code>region</code> (<code>-r</code>), <code>image-id</code> (<code>-i</code>) e <code>cluster-configuration</code> <code>/image-</code>	10 febbraio 2022

Modifica	Descrizione	Data
	configuration (-c) alla CLI AWS ParallelCluster . • Aggiunge il supporto per NEW_CHANGED_DELETE l'opzione per il parametro FSx AutoImportPolicy for Lustre. • Aggiungi parallelcluster:compute-resource-name tag alle LaunchTemplates risorse EC2 utilizzate dai nodi di calcolo. • Migliora i gruppi di sicurezza creati all'interno del cluster per consentire le connessioni in entrata da gruppi di sicurezza personalizzati quando vengono specificati SecurityGroups i parametri per alcune code dei nodi principali. and/or • Installa i driver NVIDIA e la libreria CUDA per ARM. Modifiche: • Aggiorna Slurm alla versione 21.08.5 (da 20.11.8). • Aggiorna il plugin Slurm alla versione (da) 21.08. 20.11	

Modifica	Descrizione	Data
	• Aggiorna NICE DCV alla versione 2021.3-11591 (da 2021.1-10851). • Aggiorna il driver NVIDIA alla versione 470.103.01 (da 470.57.02). • Aggiorna NVIDIA Fabric Manager alla versione 470.103.01 (da 470.57.02). • Aggiorna CUDA alla versione 11.4.4 (da 11.4.0). • Intel MPI aggiornato alla versione 2021 Update 4 (aggiornato dalla versione 2019 Update 8). Per ulteriori informazioni, vedere Intel® MPI Library 2021 Update 4. • Aggiorna PMix alla versione 3.2.3 (da 3.1.5). • Rimuovi il dumping dei nodi di calcolo guasti su. /home/logs/compute I file di log dei nodi di calcolo sono disponibili nei CloudWatch e nei log della console Amazon EC2. • Abilita la possibilità di sopprimere e allungare i validatori <code>SlurmQueues</code> . <code>ComputeResources</code>	

Modifica	Descrizione	Data
	• Disattiva l'aggiornamento del pacchetto al momento del lancio dell'istanza su Amazon Linux 2. • Disattiva i metadati ImageBuilder avanzati delle immagini di Amazon EC2 durante la creazione AWS ParallelCluster di immagini personalizzate. • Imposta esplicitamente l'<code>cloud-init</code> origine dati come EC2. Ciò consente di risparmiare tempo di avvio per le piattaforme Ubuntu e CentOS. • Usa il nome della risorsa di calcolo anziché il tipo di istanza nel nome del modello di avvio della flotta di calcolo. • Reindirizza <code>stderr</code> e <code>stdout</code> al file di registro della CLI per evitare testo indesiderato nell'output della CLI di <code>pcluster</code>. • Sposta le <code>configure/install</code> ricette in libri di cucina separati che vengono richiamati da quello principale. I punti di ingresso esistenti vengono mantenuti e sono compatibili con le versioni precedenti.	

Modifica	Descrizione	Data
	• Scarica le dipendenze della piattaforma Intel HPC durante la fase di creazione dell'AMI per evitare di contattare Internet durante la creazione del cluster. • Non - togliete il nome della risorsa di calcolo durante la configurazione dei nodi Slurm. • Non configurate le GPU in Slurm quando il driver NVIDIA non è installato. • Correggi l'autorizzazione in <code>ecs:ListContainerInstances</code> <code>BatchUserRole</code> • Corregge l'esportazione dei log del cluster quando non è specificato alcun prefisso, precedentemente esportati in un prefisso. <code>None</code> • Risolve il problema del rollback che non veniva eseguito in caso di errore di aggiornamento del cluster. • Correggi <code>ecs:ListContainerInstances</code> l'autorizzazione in <code>BatchUserRole</code> . • Correggi <code>RootVolume</code> lo schema <code>HeadNode</code> generando un errore se ne	

Modifica	Descrizione	Data
	KmsKeyId viene specificato uno non supportato. • Correggi i parametri mancanti di Amazon FSx da visualizzare nella dashboard . CloudWatch • Correggere. EfaSecurityGroupValidator In precedenza, poteva produrre falsi errori quando venivano forniti gruppi di sicurezza personalizzati e l'EFA era abilitato. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.0.3	AWS ParallelCluster rilasciata la versione 3.0.3. • Disabilita log4j - cve -2021-44228-hotpatch agent (Log4jHotPatch) su Amazon Linux 2 per evitare un potenziale peggioramento delle prestazioni. Per ulteriori informazioni, consulta l'annuncio di Amazon Linux Hotpatch per Apache Log4j. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti aws-parallelcluster e aws-parallelcluster-cookbook su GitHub	17 gennaio 2022

Modifica	Descrizione	Data
AWS ParallelCluster rilasciata la versione 3.0.2	AWS ParallelCluster rilasciata la versione 3.0.2. Aggiorna il programma di installazione di Elastic Fabric Adapter alla versione 1.14.1 • Configurazione EFA: (da) efa-config-1.9-1 efa-config-1.9 • Profilo EFA: efa-profile-1.5-1 (da) efa-profile-1.5 • Modulo EFA Kernel: efa-1.14.2 (da) efa-1.13.0 • Core RDMA: rdma-core-37.0 (da) rdma-core-35 • Libfabric: libfabric-1.13.2 (da) libfabric-1.13.0 • Apri MPI: openmpi40-aws-4.1.1-2 (nessuna modifica) GPUDirect RDMA è sempre abilitato se supportato dal tipo di istanza. L'opzione GdrSupport di configurazione non ha effetto. Per i dettagli sulle modifiche, consulta i CHANGELOG	5 novembre 2021

Modifica	Descrizione	Data
	file per i pacchetti aws-parallecluster , aws-parallecluster-cookbook e aws-parallecluster-nodesu . GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.0.1	AWS ParallelCluster rilasciata la versione 3.0.1. Strumento di migrazione per la configurazione del cluster - I clienti possono ora migrare le configurazioni dei cluster dal formato AWS ParallelCluster versione 2 al formato YAML-based AWS ParallelCluster versione 3. Per ulteriori informazioni, consulta <code>pcluster3-config-converter</code>. Il nodo principale può essere fermato Dopo aver interrotto la flotta di elaborazione, il nodo principale può essere arrestato e riavviato successivamente utilizzando la console Amazon EC2 o il comando CLI <code>stop-instances</code>. AWS <code>~/.aws/config</code> Regione predefinita letta dal file AWS - Per il comando <code>pcluster</code>, se la AWS Regione non è specificata nel file di configurazione,	27 ottobre 2021

Modifica	Descrizione	Data
	nell'ambiente o nella riga di comando, viene utilizzata la AWS Regione predefinita specificata nell'<code>region</code> impostazione nella <code>[default]</code> sezione del <code>~/.aws/config</code> file. Per i dettagli sulle modifiche , consulta i CHANGELOG file per i pacchetti <code>aws-parallelcluster</code>, <code>aws-parallelcluster-cookbook</code> e <code>aws-parallelcluster-node</code> su GitHub	

Modifica	Descrizione	Data
AWS ParallelCluster è stata rilasciata la versione 3.0.0	AWS ParallelCluster rilasciata la versione 3.0.0. Supporto per la gestione dei cluster tramite Amazon API Gateway • I clienti possono ora gestire e distribuire i cluster tramite endpoint HTTP con Amazon API Gateway. Questo apre nuove possibilità per flussi di lavoro basati su script o basati su eventi. Anche l'interfaccia a riga di AWS ParallelCluster comando (CLI) è stata riprogettata per essere compatibile con questa API e include una nuova opzione di output JSON. Questa nuova funzionalità consente ai clienti di implementare funzionalità di base simili anche utilizzando la CLI. Creazione di AMI personalizzate migliorate • I clienti ora hanno accesso a un processo più solido per la creazione e la gestione di AMI personalizzate utilizzan	10 settembre 2021

Modifica	Descrizione	Data
	do EC2 Image Builder. Le AMI personalizzate possono ora essere gestite tramite un file di AWS ParallelCluster configurazione separato e possono essere create utilizzando il comando pcluster build-image nell'interfaccia a riga di comando. AWS ParallelCluster Per i dettagli sulle modifiche, consulta i CHANGELOG file per i pacchetti aws-parallelcluster, aws-parallelcluster-cookbook e aws-parallelcluster-node su. GitHub	

PCUI

Modifica	Descrizione	Data
Rilasciata la versione 2026.04.0 di PCUI	Rilasciata la versione PCUI 2026.04.0 Ultime modifiche: • Rimuovi il supporto per AWS ParallelCluster le versioni $\leq 3.5.0$. Funzionalità: • Aggiungi il supporto per la gestione di più AWS	30 aprile 2026

Modifica	Descrizione	Data
	ParallelCluster versioni da un'unica distribuzione PCUI. • Aggiungi il supporto per PCUI nella us-gov-east-1 regione. Correzioni di bug • Risolvi un problema per garantire che le opzioni del sistema operativo nella procedura guidata di creazione del cluster riflettano i sistemi operativi supportati dalla versione selezionata. AWS ParallelCluster • Risolve un problema per consentire a PCUI di offrire l'intera gamma di tipi di istanze selezionabili nella procedura guidata. • Migliora il messaggio CloudFormation di errore quando il <code>Version</code> parametro non è specificato. Sicurezza: • Riduci il rischio di attacchi Path Traversal nella PCUI Lambda. • Risolve una vulnerabilità di iniezione di comandi del	

Modifica	Descrizione	Data
	sistema operativo nelle chiamate SSM sul cluster. SendCommand HeadNode • Risolve un bug che causava la perdita dei token di accesso nei log. CloudWatch • Richieste di aggiornamento alla versione 2.33.1 (dalla 2.32.0). • Aggiorna la crittografia alla versione 46.0.7 (dalla 44.0.1). • Aggiorna boto3 alla versione 1.42.91 (da 1.24.30). • Aggiorna urllib3 alla versione 2.6.3 (dalla 1.26.20). • Aggiorna ecdsa alla versione 0.19.2 (dalla 0.18.0). • Aggiorna certifi alla versione 2026.2.25 (dalla 2024.7.4). • Aggiorna flask-cors alla versione 6.0.2 (dalla 4.0.2). • Aggiorna python-jose alla versione 3.5.0 (dalla 3.4.0). • Aggiornamento successivo alla versione 15.5.10 (dalla 14.2.26). • Aggiorna eslint-config-next alla versione 15.5.10 (dalla 14.0.4).	

Modifica	Descrizione	Data
	• Aggiorna @ babel/runtime alla versione 7.27.0 (dalla 7.23.5). • Aggiorna brace-expansion alla versione 1.1.12 (dalla 1.1.11).	

Modifica	Descrizione	Data
Rilasciata la versione PCUI 2025.04.0	Rilasciata la versione PCUI 2025.04.0 Ultime modifiche: • Rimuovi il valore predefinito per la versione PC. Ora l'utente deve specificare la versione del PC da utilizzare. Funzionalità: • Aggiungi un nuovo parametro stack <code>'AdditionalPolicies PCAPI '</code> per aggiungere autorizzazioni personalizzate per il ruolo API ParallelCluster Lambda, oltre a quelle predefinite. Correzioni di bug • Correggi la distribuzione PCUI nelle sottoreti private facendo in modo che il modello PCUI utilizzi e restituisca gli URL corretti. • Risolve un problema che impedisce il caricamento di oltre 200 lavori nella scheda Job status. (Vedi https://github.com/aws/aws-parallelcluster-ui/issues/376).	16 aprile 2025

Modifica	Descrizione	Data
	Sicurezza: • Aggiorna Python da 3.9 a 3.12. • Aggiorna cross-spawn da 7.0.3 a 7.0.6 per risolvere la vulnerabilità. CVE-2024-21538 • Richieste di aggiornamento da 2.31.0 a 2.32.0 da indirizzare. CVE-2024-35195 • Aggiorna urllib3 da 1.26.18 a 1.26.19 all'indirizzo. CVE-2024-37891 • Aggiorna la crittografia da 42.0.4 a 44.0.1 all'indirizzo. CWE-1395 • Aggiorna certifi da 2023.7.22 a 2024.7.4 all'indirizzo. CVE-2024-39689 • Aggiorna jinja2 da 3.1.3 a 3.1.6 a address and. CVE-2024-56201CVE-2024-56326 • Aggiorna serverless_wsgi.py alla versione 3.0.5. • Aggiorna Werkzeug dalla versione 2.3.8 alla versione 3.0.6 a address e. CVE-2024-34069CVE-2024-49766CVE-2024-49767	

Modifica	Descrizione	Data
	• Aggiorna Axios dalla versione 1.6.7 alla versione 1.8.2 per indirizzare. CVE-2024-39338 • Aggiornamento Next.js dalla versione 14.1.1 alla versione 14.2.25 all'indirizzo e. CVE-2024-51479CVE-2024-46982CVE-2025-29927 • Aggiorna idna dalla 3.4 alla versione 3.7 all'indirizzo. CVE-2024-3651 • Aggiorna nanoid dalla versione 3.3.7 alla versione 3.3.8 per indirizzare. CVE-2024-55565 • Aggiorna python-jose dalla 3.3.0 alla versione 3.4.0 per indirizzare. CVE-2022-29217	

Modifica	Descrizione	Data
Rilasciata la versione PCUI 2024.11.0	Rilasciata la versione PCUI 2024.11.0 Correzioni di bug • Imposta in modo esplicito la policy per un repository privato ECR per evitare che la rimozione delle policy su un aggiornamento dello stack influisca su una funzione Lambda. La policy include le autorizzazioni richieste dalla funzione Lambda per recuperare il codice.	22 novembre 2024

Modifica	Descrizione	Data
È stata rilasciata la versione 2024.10.0 di PCUI	Rilasciata la versione PCUI 2024.10.0 Modifiche: • Aggiungi il supporto per AWS ParallelCluster 3.11.1. • Aggiungi il supporto per On-Demand Capacity Reservations e Capacity Block nella procedura guidata. • Aggiungi le famiglie g6, m7 e p5 all'elenco dei tipi di istanze supportati nella procedura guidata. • Aggiungi nuovi parametri opzionali dello stack per configurare il dominio personalizzato sia per PCUI che per Cognito. Correzioni di bug • Risolve un bug che interrompeva la configurazione del dominio personalizzato. Sicurezza: • Aggiornamento Flask-CORS da 3.0.10 a 4.0.2 per risolvere la vulnerabilità. CVE-2024-6221	22 ottobre 2024

Modifica	Descrizione	Data
	• Aggiornamento lint-staged da 13.0.3 a 15.2.5 per risolvere la vulnerabilità. CVE-2024-4068 • Changelog completo	
Rilasciata la versione PCUI 2024.05.0	Rilasciata la versione PCUI 2024.05.0. Correzioni di bug: • È stato corretto un bug nel frontend che bloccava l'interfaccia utente quando l'utente apre il pannello Job Status. • Changelog completo	14 maggio 2024
Rilasciata la versione PCUI 2024.04.0	Rilasciata la versione PCUI 2024.04.0. Funzionalità: • È stato aggiunto il supporto per la versione 3.9.1 AWS ParallelCluster • Changelog completo	17 aprile 2024

Modifica	Descrizione	Data
Rilasciata la versione PCUI 2024.03.0	Rilasciata la versione PCUI 2024.03.0. Funzionalità: • È stato aggiunto il supporto per la versione 3.9.0 AWS ParallelCluster • È stato aggiunto il supporto per Ubuntu 22.04 e Red Hat Enterprise Linux 9 • Ubuntu 18.04 obsoleto Correzioni di bug • È stato risolto il problema che impediva la visualizzazione di alcuni cluster quando si utilizzavano molti cluster Per i dettagli sulle modifiche , consulta i CHANGELOG file relativi al aws-parallelcluster-ui pacchetto su. GitHub	12 marzo 2024

Modifica	Descrizione	Data
È stata rilasciata la versione 2024.02.0 di PCUI	Rilasciata la versione PCUI 2024.02.0 Modifiche: • Aggiornato l'ambiente di runtime Lambda a Python v3.9 Per i dettagli sulle modifiche , consulta CHANGELOG i file relativi al pacchetto su. aws-parallelcluster-ui GitHub	8 febbraio 2024

Modifica	Descrizione	Data
È stata rilasciata la versione PCUI 2023.12.0	Rilasciata la versione PCUI 2023.12.0. Funzionalità: • È stato aggiunto il supporto per l'implementazione di PCUI con reti private. • È stata aggiunta la possibilità di applicare facoltativamente un limite di autorizzazioni a ogni ruolo IAM creato dalle infrastrutture PCUI e PCAPI • È stata aggiunta la possibilità di applicare facoltativamente un prefisso a ogni ruolo e policy IAM creati dall'infrastruttura PCUI e PCAPI. • È stato aggiunto il supporto per la ParallelCluster versione 3.8.0, senza parità di funzionalità nella procedura guidata. Per i dettagli sulle modifiche , consulta i CHANGELOG file per il pacchetto aws-parallelcluster-ui su. GitHub	21 dicembre 2023

Modifica	Descrizione	Data
È stata rilasciata la versione PCUI 2023.10.0	È stata rilasciata la versione PCUI 2023.10.0. Funzionalità: • È stato aggiunto il supporto per la ParallelCluster versione 3.7.2 con parità di funzionalità nella procedura guidata limitata a FSx File Cache e compatibilità della pianificazione basata sulla memoria con più tipi di istanze. Correzioni di bug • È stato risolto il problema che causava errori dell'interfaccia utente quando PCUI non dispone delle autorizzazioni per interagire con Cost Explorer. Miglioramenti • Sicurezza migliorata riducendo il TTL del token di accesso da 10 minuti a 5 minuti. Per i dettagli sulle modifiche , consulta CHANGELOG i file	20 ottobre 2023

Modifica	Descrizione	Data
	per il pacchetto aws-parallelcluster-ui su. GitHub	
È stata rilasciata la versione PCUI 2023.06.0	Rilasciata la versione PCUI 2023.06.0. Modifiche: • È stata aggiornata la versione AWS ParallelCluster API predefinita alla 3.6.0. Correzioni di bug • Risolto il problema della distribuzione non funzionante per AWS GovCloud () RegionUS-West. • Il pannello diviso ora carica correttamente i dettagli del cluster dopo l'inizio della creazione. Note: • La funzionalità di monitoraggio dei costi non è disponibile nelle regioni AWS GovCloud (Stati Uniti). Per i dettagli sulle modifiche , consulta CHANGELOG i file per il pacchetto aws-parallelcluster-ui su. GitHub	7 giugno 2023

Modifica	Descrizione	Data
È stata rilasciata la versione PCUI 2023.05.0	Rilasciata la versione PCUI 2023.05.0. Miglioramenti: • A partire dalla AWS ParallelCluster versione 3.6.0, aggiungi il supporto per RHEL 8. • Aggiungi il monitoraggio dei costi del cluster. • A partire dalla AWS ParallelCluster versione 3.6.0, aumenta le quote di risorse di coda e di calcolo. Modifiche: • È stata migliorata l'interfaccia utente della procedura guidata per la creazione del cluster. • È stata aumentata la velocità di implementazione PCUI. • È stata migliorata l'interfaccia per l'aggiunta di un nuovo utente. • Per impostazione predefinita, le code si trovano nella sottorete del nodo principale. Correzioni di bug	16 maggio 2023

Modifica	Descrizione	Data
	• Passa alla regione corretta dopo il completamento della creazione del cluster. • Correggi la visualizzazione dell'indicatore di caricamento nella funzione «Modifica cluster». • Correggi la creazione del cluster quando la Snapshot di proprietà EBS viene rimossa. Per i dettagli sulle modifiche , consulta CHANGELOG i file per il pacchetto aws-parallelcluster-ui su. GitHub	

Modifica	Descrizione	Data
È stata rilasciata la versione PCUI 2023.04.0	Rilasciata la versione PCUI 2023.04.0. Miglioramenti: • Riprogettazione della procedura guidata per la creazione di cluster. • Riprogettazione della pagina dei log del cluster. • Aggiungi un'impostazione del nome personalizzata per l'archiviazione condivisa. • Aggiungi una selezione multipla di storage quando aggiungi storage a un cluster. • Aggiungi <code>DeletionPolicy</code> il supporto per Amazon EFS e FSx for Lustre. • Aggiungi <code>ImdsSupport</code> impostazioni nella configurazione del cluster. • Aggiungi il supporto per i tipi di istanze C7. • È stato aggiunto un tutorial sul ripristino di una versione precedente del documento AWS Systems Manager. Modifiche:	17 aprile 2023

Modifica	Descrizione	Data
	• Configurazione del cluster YAML con dimensioni fino a 1 MB. • L'utente non è disconnesso a causa di un'autorizzazione con credenziali temporanee Boto3 IAM. • Opzioni multi-threading disattivate quando viene selezionata un'istanza HPC. • Rimosso, disabilita il rollback nella pagina di creazione del cluster. • All'utente viene impedito di utilizzare il PCUI fino a quando non vengono fornite le informazioni richieste. • È possibile aggiungere fino a 10 code. • Il SSM-SessionManager RunShell documento non viene sovrascritto durante l'installazione di PCUI.	
	Correzioni di bug • Correggi il link di reimpostazione della password non funzionante • Risolve il problema delete stack causato dal EcrPrivateRepository mancato vuoto	

Modifica	Descrizione	Data
	• È stato risolto il problema di inizializzazione della casella di controllo Genera chiavi SSH nella sezione Proprietà di gestione di più utenti. • È stato risolto il problema di arresto anomalo causato da un lavoro con proprietà non definite. • Impostazioni SCRATCH FSx fisse. • Risolto il problema con il pulsante Start e Stop delle istanze, ancora abilitato dopo essere stato premuto una volta. Per i dettagli delle modifiche , consulta CHANGELOG i file per il pacchetto aws-parallelcluster-ui su. GitHub	

Terraform

Modifica	Descrizione	Data
AWS ParallelCluster Rilasciato Terraform Provider per 1.1.0	Correzioni di bug • È stato risolto un problema che causava l'errore di applicazione di terraform quando l' ParallelCluster API 3.11.x veniva utilizzata per	6 dicembre 2024

Modifica	Descrizione	Data
	distribuire cluster con nodi di accesso.	
AWS ParallelCluster Rilasciato il modulo Terraform per 1.1.0	Modifiche: • Usa AWS ParallelCluster Terraform Provider 1.x in tutti gli esempi di moduli. • Usa l' ParallelCluster API 3.11.1 in tutti gli esempi con l'API stack name. ParallelCluster • Implementa i nodi di accesso in tutti gli esempi di moduli.	6 dicembre 2024
Rilasciato Terraform Provider per 1.0.0 AWS ParallelCluster	Funzionalità: • Changelog completo	26 giugno 2024
Rilasciato il modulo Terraform per 1.0.0 AWS ParallelCluster	Funzionalità: • Changelog completo	26 giugno 2024

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.