



AWS ParallelCluster Guia do usuário (v3)

AWS ParallelCluster



AWS ParallelCluster: AWS ParallelCluster Guia do usuário (v3)

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é AWS ParallelCluster?	1
Preços	1
Conf AWS ParallelCluster configuração	2
Configurando um Conta da AWS	2
Cadastrar-se em uma Conta da AWS	2
Crie um usuário administrador	3
Criar um par de chaves	4
Instalando a AWS ParallelCluster CLI	4
Instalar AWS ParallelCluster em um ambiente virtual (recomendado)	5
Instalando AWS ParallelCluster em um ambiente não virtual usando pip	7
Instale AWS ParallelCluster como um aplicativo independente	8
Etapas a serem realizadas após a instalação	9
Instalando a interface do usuário do AWS ParallelCluster	10
Instalação da interface do usuário do AWS ParallelCluster	11
Criar domínio personalizado	13
Opções de login do grupo de usuários do Amazon Cognito	16
Identifique a versão do AWS ParallelCluster e da interface do usuário do AWS ParallelCluster.	19
Atualizar a interface de usuário do AWS ParallelCluster para uma nova versão do AWS ParallelCluster	19
Custos de interface de usuário do AWS ParallelCluster	19
Conceitos básicos	20
Configurar e criar um cluster com a AWS ParallelCluster CLI	20
Configurar e criar um cluster com a AWS ParallelCluster interface do usuário	30
Conectar-se a um cluster	32
Acesso de vários usuários aos clusters	33
Crie um Active Directory	34
Crie um cluster com um domínio AD	34
Faça login em um cluster integrado a um domínio do AD	37
Executando um trabalho de MPI	38
Exemplo de AWS Managed Microsoft AD sobre configurações de cluster LDAP(S)	39
Práticas recomendadas	43
Práticas recomendadas: seleção do tipo de instância do nó principal	43
Práticas recomendadas: desempenho da rede	43

Práticas recomendadas: alertas de orçamento	45
Práticas recomendadas: mover um cluster para uma nova versão de patch ou secundária do AWS ParallelCluster	45
Passando do AWS ParallelCluster 2.x para 3.x	47
Ações de bootstrap personalizadas	47
AWS ParallelCluster 2.x e 3.x usam sintaxe de arquivo de configuração diferente	47
Linguagem inclusiva	54
Compatibilidade com programadores	54
CLI do AWS ParallelCluster	54
Atualização da configuração do IMDS	57
Regiões compatíveis do AWS ParallelCluster	57
Usando AWS ParallelCluster	59
Interface do usuário AWS ParallelCluster	59
Configuração de VPC AWS Lambda no AWS ParallelCluster	62
AWS Identity and Access Management permissões em AWS ParallelCluster	63
AWS ParallelCluster Funções de instância do EC2	64
AWS ParallelCluster exemplo de políticas <code>pcluster</code> de usuário	65
AWS ParallelCluster políticas de exemplo de usuário para gerenciar recursos do IAM	79
AWS ParallelCluster parâmetros de configuração para gerenciar permissões do IAM	86
Configurações de rede	101
O AWS ParallelCluster em uma única sub-rede pública	102
O AWS ParallelCluster com duas sub-redes	104
O AWS ParallelCluster em uma única sub-rede privada conectado usando o AWS Direct Connect	105
AWS ParallelCluster com programador AWS Batch	106
AWS ParallelCluster em uma sub-rede única sem acesso à internet	108
Nós de login	114
Ações de bootstrap personalizadas	118
Configuração	120
Argumentos	124
Exemplo de cluster com ações de bootstrap personalizadas	124
Exemplo de atualização de um script de bootstrap personalizado para IMDSv2	126
Exemplo de atualização de uma configuração para IMDSv1	127
Como trabalhar com o Amazon S3	128
Exemplos	128
Trabalho com Instâncias spot	129

Cenário 1: Uma instância spot sem trabalhos em execução é interrompida	129
Cenário 2: Uma instância spot que executa trabalhos de nó único é interrompida	130
Cenário 3: Uma instância spot que executa trabalhos de vários nós é interrompida	130
Agendadores suportados por AWS ParallelCluster	130
Slurm Workload Manager	131
AWS Batch	195
Armazenamento compartilhado	203
Configurar o armazenamento compartilhado	206
Trabalhando com armazenamento compartilhado	209
Cotas	213
Marcação	214
Monitoramento do AWS ParallelCluster e de logs	217
Integração com Amazon CloudWatch Logs	219
Painel do Amazon CloudWatch	222
Alarmes do Amazon CloudWatch para métricas de cluster	224
Rotação de log configurado do AWS ParallelCluster	227
Logs da CLI <code>pcluster</code>	228
Logs de saída do console EC2	229
Recupere registros de UI AWS ParallelCluster e tempo de execução AWS ParallelCluster ..	230
Recuperando e preservando logs	232
AWS CloudFormation recurso personalizado	235
Pilha de provedores hospedada por AWS ParallelCluster	236
Recurso do cluster	237
Operações de cluster	241
Pilhas de solução de problemas que incluem o recurso AWS ParallelCluster personalizado	241
Elastic Fabric Adapter	242
Habilitar o Intel MPI	243
API AWS ParallelCluster	244
Documentação da API AWS ParallelCluster	244
Implantação com AWS CLI	245
Atualização da API	248
Como invocar a API AWS ParallelCluster	249
Acessando logs e métricas das APIs	251
Conecte-se ao nó principal por meio do NICE DCV	252
Certificado HTTPS do NICE DCV	252

Licenciamento do NICE DCV	252
Usar o <code>pcluster update-cluster</code>	253
Política de atualização: definições	253
Exemplos do <code>pcluster update-cluster</code>	257
AWS ParallelCluster Personalização da AMI	259
AWS ParallelCluster Considerações sobre a personalização da AMI	260
Execute testes personalizados de validação de componentes	260
Monitore o processo do Image Builder com comandos <code>pcluster</code> para auxiliar na depuração	261
Outras considerações	262
Iniciar instâncias com ODCR (On-Demand Capacity Reservations)	262
Usando o ODCR com o AWS ParallelCluster	263
Aplicação de patches de AMI e substituição de instâncias do EC2	272
Atualização ou substituição da instância do nó principal	273
Salvar dados de unidades efêmeras	273
Interromper e iniciar o nó principal de um cluster	274
Sistema operacional	276
Considerações sobre o sistema operacional	276
Referência para AWS ParallelCluster	278
Comandos CLI do AWS ParallelCluster versão 3	278
<code>pcluster</code>	279
<code>pcluster3-config-convert</code>	323
Arquivos de configuração	324
Arquivo de configuração do cluster	324
Arquivos de configuração de imagem de compilação	457
Referência da API do AWS ParallelCluster	466
<code>buildImage</code>	466
<code>createCluster</code>	471
<code>deleteCluster</code>	477
<code>deleteClusterInstances</code>	480
<code>deleteImage</code>	482
<code>describeCluster</code>	485
<code>describeClusterInstances</code>	493
<code>describeComputeFleet</code>	496
<code>describeImage</code>	498
<code>getClusterLogEvents</code>	505

getClusterStackEvents	509
getImageLogEvents	513
getImageStackEvents	517
listClusters	521
listClusterLogStreams	525
listImageLogStreams	529
listImages	533
listOfficialImages	536
updateCluster	539
updateComputeFleet	546
API da biblioteca Python do AWS ParallelCluster	548
Autorização da biblioteca Python do AWS ParallelCluster	548
Como instalar a biblioteca Python do AWS ParallelCluster	549
Operações de API do cluster	549
Operações da API da frota de computação	553
Operações de cluster e log de pilha	555
Operações da API Image	557
Operações de log de imagens e pilhas	560
Exemplo	563
AWS Lambda para a biblioteca Python do AWS ParallelCluster	564
Como funciona o AWS ParallelCluster	566
AWS ParallelCluster processos	566
clustermgtd	566
clusterstatusmgtd	567
computemgtd	568
Serviços AWS usados por AWS ParallelCluster	568
Amazon API Gateway	569
AWS Batch	569
AWS CloudFormation	569
Amazon CloudWatch	570
CloudWatch Eventos da Amazon	570
CloudWatch Registros da Amazon	570
AWS CodeBuild	571
Amazon DynamoDB	571
Amazon Elastic Block Store	571
Amazon Elastic Compute Cloud	571

Amazon Elastic Container Registry	572
Amazon EFS	572
Amazon FSx para Lustre	572
Amazon FSx para ONTAP NetApp	573
Amazon FSx para OpenZFS	573
AWS Identity and Access Management	573
AWS Lambda	574
Amazon RDS	574
Amazon Route 53	574
Amazon Simple Notification Service	574
Amazon Simple Storage Service	574
Amazon VPC	575
Elastic Fabric Adapter	575
EC2 Image Builder	575
NICE DCV	575
AWS ParallelClusterDiretórios internos	576
Tutoriais	577
Como executar o seu primeiro trabalho no AWS ParallelCluster	577
Verificar a instalação	578
Criação de seu primeiro cluster	578
Fazendo login em seu nó de cabeçalho.	579
Executar o primeiro trabalho usando o Slurm	580
Criar uma AMI personalizada do AWS ParallelCluster	581
Como personalizar a AMI do AWS ParallelCluster	582
Criar uma AMI personalizada do AWS ParallelCluster	582
Modificar uma AMI do AWS ParallelCluster	590
Integração do Active Directory	592
Configurando a criptografia de armazenamento compartilhado com uma chave AWS KMS	623
Crie a política do	625
Configurar e criar o cluster	625
Executando trabalhos em um cluster de modo de várias filas	627
Configure seu cluster	628
Crie seu cluster do	629
Faça o login no nó principal.	630
Executar trabalho no modo de várias filas	631
Uso da API AWS ParallelCluster	635

Criação de um cluster com contabilidade Slurm	649
Etapa 1: criar a VPC e as sub-redes para AWS ParallelCluster	650
Etapa 2: criar a pilha de banco de dados	650
Etapa 3: criar um cluster com a contabilidade Slurm ativada	651
Revertendo para uma versão anterior do documento do AWS Systems Manager	651
Reverter para uma versão anterior do documento SSM	652
Criação de um cluster com AWS CloudFormation	654
Criação de clusters com uma pilha de criação CloudFormation rápida	655
Criação de clusters com a interface de linha de AWS CloudFormation comando (CLI)	657
Exibir saída CloudFormation do cluster	659
Acessar seu cluster	660
Limpeza	660
AWS ParallelClusterIntegração da interface do usuário com o Identity Center	661
Habilitar o IAM Identity Center	661
Adicionar seu aplicativo ao IAM Identity Center	664
Solução de problemas do AWS ParallelCluster	672
Tentando criar um cluster	673
failureCode é OnNodeConfiguredExecutionFailure	673
failureCode é OnNodeConfiguredDownloadFailure	673
failureCode é OnNodeConfiguredFailure	674
failureCode é OnNodeStartExecutionFailure	674
failureCode é OnNodeStartDownloadFailure	675
failureCode é OnNodeStartFailure	675
failureCode é EbsMountFailure	675
failureCode é EfsMountFailure	676
failureCode é FsxMountFailure	676
failureCode é RaidMountFailure	676
failureCode é AmiVersionMismatch	677
failureCode é InvalidAmi	677
failureCode é uma HeadNodeBootstrapFailure com failureReason Falha na configuração do nó principal.	677
failureCode é uma HeadNodeBootstrapFailure com failureReason de tempo limite de criação do cluster.	678
failureCode é uma HeadNodeBootstrapFailure com failureReason de Falha no bootstrap do nó principal.	679
failureCode é ResourceCreationFailure	679

failureCode é ClusterCreationFailure	680
Vendo WaitCondition timed out... na CloudFormation pilha	680
Vendo Resource creation cancelled na CloudFormation pilha	680
Visualizando Failed to run cfn-init... ou outros erros na pilha AWS CloudFormation	680
Visualizando chef-client.log que termina com INFO: Waiting for static fleet capacity provisioning	680
Vendo Failed to run preinstall or postinstall in cfn-init.log	681
Vendo This AMI was created with xxx, but is trying to be used with xxx... na CloudFormation pilha	681
Vendo This AMI was not baked by AWS ParallelCluster... na CloudFormation pilha	681
Vendo que o comando pcluster create-cluster falha ao ser executado localmente .	681
Suporte adicional	681
Tentar executar um trabalho	682
O trabalho interativo do srun falha com erro srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf	682
Trabalho está preso no estado CF com o comando squeue	682
Executando trabalhos em grande escala e vendo nfsd: too many open connections, consider increasing the number of threads in /var/log/messages	683
Executando um trabalho de MPI	683
Tentando atualizar um cluster	684
O comando pcluster update-cluster falha ao ser executado localmente	684
Ver clusterStatus é UPDATE_FAILED com comando pcluster describe-cluster	684
A atualização do cluster atingiu o tempo limite	685
Tentar acessar o armazenamento	685
Usar um sistema de arquivos externo do Amazon FSx for Lustre	685
Usar um sistema de arquivos externo do Amazon Elastic File System	685
Tentando excluir um cluster	685
O comando pcluster delete-cluster falha ao ser executado localmente	685
A pilha de clusters não consegue ser excluída	685
Tentando atualizar a pilha de APIs AWS ParallelCluster	686
Vendo erros nas inicializações dos nós de computação	686
Vendo Node bootstrap error em clustermgtd.log	686

Eu configurei reservas de capacidade sob demanda (ODCRs) ou instâncias reservadas zonais	686
Ver An error occurred (VcpuLimitExceeded) no <code>slurm_resume.log</code> quando não consegui executar um trabalho ou em <code>clustermgtd.log</code> quando eu não consigo criar um cluster	688
Ver An error occurred (InsufficientInstanceCapacity) no <code>slurm_resume.log</code> quando não consegui executar um trabalho ou em <code>clustermgtd.log</code> quando eu não consigo criar um cluster	688
Vendo que os nós estão em estado DOWN com Reason (Code:InsufficientInstanceCapacity)... ..	688
Vendo cannot change locale (en_US.utf-8) because it has an invalid name em <code>slurm_resume.log</code>	688
Nenhum dos cenários anteriores se aplica à minha situação	689
Métricas de integridade do cluster para solução de problemas	689
Visualizando o gráfico de Instance Provisioning Errors (Erros de provisionamento de instâncias)	690
Visualizando o gráfico de Unhealthy Instance Errors (Erros de instância não saudáveis)	692
Visualizando o gráfico de Tempo de inatividade da frota de computadores	694
Solução de problemas de implantação de cluster	694
Veja os eventos AWS CloudFormation em CREATE_FAILED	695
Use a CLI para visualizar fluxos de log	697
Recrie o cluster com falha com <code>rollback-on-failure</code>	699
Solucionar problemas de escala	700
Logs principais para depuração	701
Ver o erro <code>InsufficientInstanceCapacity</code> no <code>slurm_resume.log</code> quando não consegui executar um trabalho ou em <code>clustermgtd.log</code> quando eu não consigo criar um cluster	688
Solução de problemas de inicialização do nó	704
Solução de problemas inesperados de substituições e encerramentos de nós	706
Substituindo, encerrando ou desligando instâncias e nós problemáticos	708
Status <code>Inactive</code> da fila (partição)	708
Solucionando outros problemas conhecidos de nós e tarefas	708
Grupos de posicionamento e problemas de execução de instâncias	708
Diretórios que não podem ser substituídos	709
Solucionar problemas no NICE DCV	709
Logs para NICE DCV	709

Problemas com o Ubuntu NICE DCV	710
Solução de problemas em clusters com integração AWS Batch	710
Problemas no nó principal	711
Problemas de computação	711
Falhas de trabalhos	711
Erro de tempo limite de conexão no URL do endpoint	711
Solução de problemas de integração de vários usuários com o Active Directory	711
Solução de problemas específicos do Active Directory	712
Habilitar modo de depuração	713
Como passar do LDAPS para o LDAP	713
Como desabilitar a verificação do certificado do servidor LDAPS	714
Como fazer login com uma chave SSH em vez de uma senha	714
Como redefinir uma senha de usuário e senhas expiradas	714
Como verificar o domínio associado	715
Como solucionar problemas com certificados	716
Como verificar se a integração com o Active Directory está funcionando	718
Como solucionar problemas de login em nós de computação	718
Problemas conhecidos com trabalhos SimCenter StarCCM+ em um ambiente multiusuário	719
Problemas conhecidos com a resolução do nome de usuário	719
Como resolver problemas de criação do diretório inicial	720
Solução de problemas de AMI personalizada	721
Solução de problemas de tempo limite de atualização de cluster quando cfn-hup não está em execução	722
Solução de problemas de rede	722
Problemas de cluster em uma única sub-rede pública	722
Falha na atualização do cluster na ação personalizada onNodeUpdated	723
Vendo erros com a configuração personalizada Slurm	723
Alarmes de cluster	724
Suporte adicional	724
AWS ParallelCluster política de suporte	725
Segurança	726
Informações de segurança para serviços usados pela AWS ParallelCluster	727
Proteção de dados	727
Criptografia de dados	728
Consulte também	730

Gerenciamento de identidade e acesso	730
Validação de conformidade	731
Impor o TLS 1.2	732
Determinar os protocolos atualmente compatíveis	732
Compilar OpenSSL e Python	734
Notas de release e histórico de documentos	736
.....	dcccxxxi

O que é AWS ParallelCluster?

AWS ParallelCluster é uma ferramenta de gerenciamento de cluster de código aberto compatível com a AWS que ajuda você a implantar e gerenciar clusters de computação de alta performance (HPC) na Nuvem AWS.. Ele configura automaticamente os recursos de computação, agendador e o sistema de arquivos compartilhado necessários. Você pode usar AWS ParallelCluster com os agendadores AWS Batch e Slurm.

Com AWS ParallelCluster, você pode criar e implantar rapidamente ambientes computacionais de HPC de prova de conceito e produção. Também é possível criar e implantar fluxos de trabalho de nível superior no topo do AWS ParallelCluster, como um portal genômico que automatiza todo o fluxo de trabalho de sequenciamento de DNA.

Você pode acessar o AWS ParallelCluster usando qualquer um dos seguintes métodos:

- [AWS ParallelCluster command line interface \(CLI\)](#)
- [API do AWS ParallelCluster](#)
- [AWS ParallelClusterUI](#) (adicionada com a versão 3.5.0)
- [API da biblioteca Python do AWS ParallelCluster](#) (adicionada com a versão 3.5.0)
- Como um [AWS CloudFormation recurso personalizado](#) (adicionada com a versão 3.6.0)

Preços

Ao usar a interface de linha de AWS ParallelCluster comando (CLI) ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Conf AWS ParallelCluster igituração

Tópicos

- [Configurando um Conta da AWS](#)
- [Criar um par de chaves](#)
- [Instalando a interface de linha de AWS ParallelCluster comando \(CLI\)](#)
- [Etapas a serem realizadas após a instalação](#)
- [Instalando a interface do usuário do AWS ParallelCluster](#)
- [Começando com AWS ParallelCluster](#)
- [Acesso de vários usuários aos clusters](#)
- [Práticas recomendadas](#)
- [Passando do AWS ParallelCluster 2.x para 3.x](#)
- [Regiões compatíveis do AWS ParallelCluster](#)

Configurando um Conta da AWS

Configure uma conta da AWS para usar o AWS ParallelCluster.

Cadastrar-se em uma Conta da AWS

Se você ainda não tem uma Conta da AWS, siga as etapas a seguir para criar uma.

Para se cadastrar em uma Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se cadastra em uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, [atribua acesso administrativo a um usuário administrativo](#) e use somente o usuário raiz para realizar as [tarefas que exigem acesso do usuário raiz](#).

A AWS envia um e-mail de confirmação depois que o processo de cadastramento é concluído. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

Crie um usuário administrador

Depois de se inscrever em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS com o AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteger seu Usuário raiz da conta da AWS

1. Faça login no [AWS Management Console](#) como o proprietário da conta ao escolher a opção Usuário raiz e inserir o endereço de e-mail da Conta da AWS. Na próxima página, digite sua senha.

Para obter ajuda ao fazer login usando o usuário raiz, consulte [Fazer login como usuário raiz](#) no Guia do usuário do Início de Sessão da AWS.

2. Ative a autenticação multifator (MFA) para o usuário raiz.

Para obter instruções, consulte [Habilitar um dispositivo MFA virtual para o usuário raiz de sua conta da Conta da AWS para seu \(console\)](#) no Guia do usuário do IAM.

Criar um usuário administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte [Ativação AWS IAM Identity Center](#) no Guia AWS IAM Identity Center do usuário.

2. No IAM Identity Center, conceda acesso administrativo a um usuário administrativo.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Login como usuário administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

Para obter ajuda com o login utilizando um usuário do Centro de Identidade do IAM, consulte [Fazer login no portal de acesso da AWS](#), no Guia do usuário do Início de Sessão da AWS.

Criar um par de chaves

Para implantar clusters, AWS ParallelCluster inicie instâncias do EC2 para criar o nó principal do cluster e os nós de computação. Para realizar tarefas de cluster, como executar e monitorar trabalhos ou gerenciar usuários, você deve acessar o nó principal do cluster. Para verificar se você pode acessar a instância do nó principal usando SSH, você deve usar um par de chaves EC2. Para saber como criar um par de chaves, consulte [Criar um par de chaves](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias do Linux.

Instalando a interface de linha de AWS ParallelCluster comando (CLI)

AWS ParallelCluster é distribuído como um pacote Python e instalado usando o gerenciador de pacotes `pip` Python. Para ver instruções sobre como instalar pacotes Python, consulte [Instalar pacotes](#) no Guia do usuário de pacotes do Python.

Formas de instalar AWS ParallelCluster:

- [Instalar AWS ParallelCluster em um ambiente virtual \(recomendado\)](#)
- [Instalando AWS ParallelCluster em um ambiente não virtual usando pip](#)
- [Instale AWS ParallelCluster como um aplicativo independente](#)

Você pode encontrar o número da versão da CLI mais recente na [página de lançamentos](#) em GitHub. Neste guia, os exemplos de comando pressupõem que você tenha instalado uma versão do Python posterior à versão 3.6. Os exemplos do comando `pip` usam a versão `pip3`.

Gerencie AWS ParallelCluster 2 e AWS ParallelCluster 3

Para clientes que usam AWS ParallelCluster 2 e AWS ParallelCluster 3 e desejam gerenciar as CLIs dos dois pacotes, recomendamos que você instale AWS ParallelCluster 2 e AWS ParallelCluster 3 em [ambientes virtuais](#) diferentes. Isso garante que você possa continuar usando cada versão do AWS ParallelCluster e quaisquer recursos de cluster associados.

Instalar AWS ParallelCluster em um ambiente virtual (recomendado)

Recomendamos que você instale AWS ParallelCluster em um ambiente virtual para evitar conflitos de versão de requisitos com outros pip pacotes.

Pré-requisitos

- AWS ParallelCluster requer o Python 3.7 ou posterior. Se você ainda não o tiver instalado, [baixe uma versão compatível](#) para sua plataforma em [python.org](#).

Para instalar AWS ParallelCluster em um ambiente virtual

1. Se o `virtualenv` não estiver instalado, instale o `virtualenv` usando `pip3`. Se `python3 -m virtualenv help` exibir informações de ajuda, vá para a etapa 2.

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

Execute `exit` para sair da janela do terminal atual e abrir uma nova janela de terminal para selecionar as alterações no ambiente.

2. Crie um ambiente virtual e nomeie-o.

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

Como alternativa, você pode usar a opção `-p` para especificar uma versão do Python.

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

3. Ative seu novo ambiente virtual.

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

4. Instale AWS ParallelCluster em seu ambiente virtual.

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

5. Instale o Node Version Manager e a versão mais recente do Node.js do Long-Term Support (LTS). AWS Cloud Development Kit (AWS CDK) (AWS CDK) requer Node.js CloudFormation para geração de modelos.

Note

Se a instalação do Node.js não estiver funcionando na sua plataforma, você poderá instalar uma versão LTS antes da versão mais recente do LTS. Para obter mais informações, consulte o [cronograma de lançamento do Node.js](#) e os [pré-requisitos do AWS CDK](#).

Exemplo de comando de instalação do Node.js:

```
$ 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. Verifique se AWS ParallelCluster está instalado corretamente.

```
$ pcluster version
{
  "version": "3.7.0"
}
```

Use o comando `deactivate` para sair do ambiente virtual. Toda vez que iniciar uma sessão, é necessário [reativar o ambiente](#).

Para atualizar para a versão mais recente do AWS ParallelCluster, execute o comando de instalação novamente.

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

Instalando AWS ParallelCluster em um ambiente não virtual usando pip

Pré-requisitos

- AWS ParallelCluster requer o Python 3.7 ou posterior. Se você ainda não o tiver instalado, [baixe uma versão compatível](#) para sua plataforma em [python.org](#).

Instalar AWS ParallelCluster

1. Use pip para instalar AWS ParallelCluster.

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

Quando você usa o `--user` switch, pip AWS ParallelCluster instala `~/.local/bin` em.

2. Instale o Node Version Manager e a versão mais recente do Node.js do Long-Term Support (LTS). AWS Cloud Development Kit (AWS CDK) (AWS CDK) requer Node.js CloudFormation para geração de modelos.

Note

Se a instalação do Node.js não estiver funcionando na sua plataforma, você poderá instalar uma versão LTS antes da versão mais recente do LTS. Para obter mais informações, consulte o [cronograma de lançamento do Node.js](#) e os [pré-requisitos do AWS CDK](#).

```
$ nvm install --lts=Gallium
```

```
$ 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. Verifique se AWS ParallelCluster está instalado corretamente.

```
$ pcluster version
{
```

```
"version": "3.7.0"  
}
```

4. Para atualizar para a versão mais recente, execute o comando de instalação novamente.

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

Instale AWS ParallelCluster como um aplicativo independente

Instale AWS ParallelCluster como um aplicativo independente em seu ambiente. Siga as instruções para instalação AWS ParallelCluster em um sistema operacional disponível na seção a seguir.

Pré-requisitos

- Um ambiente com um sistema operacional compatível com uma versão disponível do instalador.

Note

AWS ParallelCluster requer NodeJS. O instalador inclui uma versão em pacote do NodeJS (v18), que é instalada se ainda não existir. Se o seu sistema não for compatível com o NodeJS v18, você deve instalar o NodeJS antes de instalar AWS ParallelCluster.

Linux

Linux x86 (64-bit)

Instale AWS ParallelCluster em seu ambiente.

1. Baixe o [instalador do pcluster](#) mais recente.
2. Descompacte o pacote do instalador e instale AWS ParallelCluster usando os seguintes comandos:

```
$ unzip pcluster-installer-bundle-3.9.1.608-node-v18.17.1-Linux_x86_64-signed.zip  
-d pcluster-installer-bundle  
$ cd pcluster-installer-bundle  
$ chmod +x install_pcluster.sh
```

3. Execute o script de instalação a seguir.

```
$ bash install_pcluster.sh
```

4. Verifique se AWS ParallelCluster está instalado corretamente.

```
$ pcluster version  
{  
  "version": "3.9.1"  
}
```

Solução de problemas de erros de instalação do **pcluster**

- Se a AWS ParallelCluster versão não for retornada na etapa 4, reinicie source o `bash_profile` terminal ou atualize a `PATH` variável para incluir o novo diretório binário, conforme mostrado no exemplo a seguir:

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

- Se você usar sua instalação do `pcluster` para criar clusters especificados com `CustomActions` como recursos HTTPS, em vez de URIs do S3, talvez veja uma mensagem `WARNING` indicando que esses recursos podem não ser verificados (`[SSL : CERTIFICATE_VERIFY_FAILED]`). Isso é causado por um problema conhecido e você pode ignorar esse aviso se confiar na autenticidade dos recursos especificados.

Versões anteriores do pacote de instalação

- Nenhum

Etapas a serem realizadas após a instalação

Você pode verificar se AWS ParallelCluster está instalado corretamente executando [pcluster version](#).

```
$ pcluster version  
{  
  "version": "3.7.0"  
}
```

AWS ParallelCluster é atualizado regularmente. Para atualizar para a versão mais recente do AWS ParallelCluster, execute o comando de instalação novamente. Para obter mais informações sobre a versão mais recente do AWS ParallelCluster, consulte as [notas AWS ParallelCluster de lançamento](#).

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

Para desinstalar AWS ParallelCluster, use `pip3 uninstall`.

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

Se não tem o Python nem o `pip3`, use o procedimento para o seu ambiente.

Instalando a interface do usuário do AWS ParallelCluster

A interface do usuário do AWS ParallelCluster é uma interface baseada na Web que espelha a CLI `pccluster` do AWS ParallelCluster, ao mesmo tempo em que fornece uma experiência semelhante à de um console. Você instala e acessa a interface do usuário do AWS ParallelCluster na sua Conta da AWS. Quando você a executa, a interface do usuário do AWS ParallelCluster acessa uma instância da API AWS ParallelCluster hospedada no Amazon API Gateway em sua Conta da AWS. Para obter mais informações sobre a interface do usuário do AWS ParallelCluster, consulte [Interface do usuário AWS ParallelCluster](#).

Pré-requisitos:

- Um Conta da AWS
- Acesso ao AWS Management Console

Tópicos

- [Instalação da interface do usuário do AWS ParallelCluster](#)
- [Criar domínio personalizado](#)
- [Opções de login do grupo de usuários do Amazon Cognito](#)
- [Identifique a versão do AWS ParallelCluster e da interface do usuário do AWS ParallelCluster.](#)
- [Atualizar a interface de usuário do AWS ParallelCluster para uma nova versão do AWS ParallelCluster](#)
- [Custos de interface de usuário do AWS ParallelCluster](#)

Instalação da interface do usuário do AWS ParallelCluster

Para instalar uma instância da interface do usuário do AWS ParallelCluster, escolha um link de criação rápida AWS CloudFormation para a Região da AWS na qual você criará os clusters. O URL de criação rápida leva você a um Assistente de criação de pilha, onde você pode fornecer entradas de criação rápida de modelo de pilha e pode implantar a pilha. Para obter mais informações sobre pilhas de CloudFormation criação rápida, consulte [Criação de links de criação rápida para pilhas](#) no Guia do usuário. AWS CloudFormation

Note

Você só pode criar e editar clusters ou criar imagens com a mesma versão do AWS ParallelCluster usada para instalar a interface de usuário do AWS ParallelCluster.

Links de criação rápida da interface do usuário do AWS ParallelCluster por região

Links de criação rápida para interface do usuário

[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](#)

Links de criação rápida para interface do usuário

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

[ap-southeast-1](#)

[ap-southeast-2](#)

[us-gov-west-1](#)

Use um link de criação rápida AWS CloudFormation para implantar uma pilha de interface de usuário AWS ParallelCluster com pilhas aninhadas do Amazon Cognito, do API Gateway e do Amazon EC2 Systems Manager.

1. Faça login no AWS Management Console.
2. Implante a interface do usuário AWS ParallelCluster escolhendo um link de criação rápida Região da AWS da tabela no início desta seção. Isso leva você ao CloudFormation Create Stack Wizard no console.
3. Insira um endereço de e-mail válido para o e-mail do administrador.

Depois que a implantação for concluída com êxito, a interface do usuário do AWS ParallelCluster enviará uma senha temporária para esse endereço de e-mail. Use a senha temporária para acessar a interface de usuário do AWS ParallelCluster. Se você excluir o e-mail antes de salvar ou usar a senha temporária, deverá excluir a pilha e reinstalar a interface de usuário do AWS ParallelCluster.

4. Mantenha o resto do formulário em branco ou insira valores para parâmetros (opcionais) para personalizar a construção da interface de usuário do AWS ParallelCluster.
5. Anote o nome da pilha para usá-lo em etapas posteriores.
6. Navegue até Recursos. Concorde com os CloudFormation recursos.
7. Escolha Create. Leva cerca de 15 minutos para que a implantação da API AWS ParallelCluster e da interface do usuário AWS ParallelCluster seja concluída.

8. Visualize os detalhes da pilha à medida que a pilha é criada.
9. Depois que a implantação for concluída, abra o e-mail do administrador que foi enviado para o endereço que você inseriu. Ele contém uma senha temporária que você deve usar para acessar a interface de usuário do AWS ParallelCluster. Se você excluir permanentemente o e-mail e ainda não tiver feito login na interface do usuário do AWS ParallelCluster, deverá excluir a pilha de interface do usuário do AWS ParallelCluster que você criou e reinstalar a interface do usuário do AWS ParallelCluster.
10. Na lista de pilhas do console AWS CloudFormation, escolha o link para o nome da pilha que você anotou em uma etapa anterior.
11. Em Detalhes da pilha, escolha Resultados e selecione o link para a chave denominada **Stackname** URL para abrir a interface do usuário do AWS ParallelCluster. **Stackname** será o nome que você anotou na etapa anterior.
12. Insira a senha temporária. Siga as etapas para criar sua própria senha e fazer login.
13. Agora você está na página inicial da interface do usuário do AWS ParallelCluster na Região da AWS que você selecionou.
14. Para começar a usar a interface de usuário do AWS ParallelCluster, consulte [Configurar e criar um cluster com a interface do usuário do AWS ParallelCluster](#).

Note

As sessões do PCUI têm uma duração padrão de 5 minutos, que é o valor mínimo fornecido pelo Cognito a partir do PCUI 2023.12.0. Portanto, espera-se que um usuário removido dos Grupos de Usuários do Cognito ainda possa acessar o sistema até que a sessão expire.

Criar domínio personalizado

Saiba como criar um domínio personalizado para a interface do usuário do AWS ParallelCluster. A interface do usuário fica hospedada no Amazon API Gateway em sua Conta da AWS. Como criar um nome de domínio personalizado usando o console do API Gateway

Pré-requisitos:

- Você ter um Conta da AWS.
- Você ter uma instância de interface de usuário do AWS ParallelCluster que você possa acessar.
- Você possuir um domínio.

- Você poder alterar configurações básicas do Sistema de Nomes de Domínio (DNS).

Etapa 1: Criar um domínio no Amazon API Gateway

1. No AWS Management Console, navegue até o [API Gateway](#), onde você estará listada sua API de interface de usuário do AWS ParallelCluster.
2. No painel de navegação, selecione Nomes de domínio personalizados.
3. Escolha Create.
4. Em Detalhes do domínio, insira seu nome de domínio.
5. Na configuração do Endpoint, escolha um certificado ACM existente ou escolha Criar um novo certificado ACM.

(Opcional) Crie um certificado

- a. No console do ACM, escolha Solicitar.
- b. Em Nomes de domínio, insira um nome para o domínio.
- c. Em Método de validação, escolha um método de validação.

Se você escolher Validação por e-mail, um e-mail será enviado para o endereço de e-mail cadastrado no registrador do domínio.

- d. Selecione Eu aprovo para ativar o certificado.

Etapa 2: Configurar o mapeamento de API

1. Em [API Gateway](#), Nomes de domínio personalizados your-domain-name, escolha Configurar mapeamentos de API.
2. Escolha Nomes de domínios personalizados.
3. Escolha Adicionar novo mapeamento.
4. Escolha a API de interface do usuário do AWS ParallelCluster, o Estágio \$default e Salvar.
5. No Nome de domínio do API Gateway, copie o valor que será usado nas próximas etapas.

Etapa 3: Configurar o DNS

- Crie uma regra DNS CNAME que direcione seu domínio para o domínio do API Gateway. Insira somente o domínio. Por exemplo, não adicione o estágio, como beta ou prod. *Substitua*

abcde12345 pelo ID da API do API Gateway e substitua *us-east-2* pela Região da AWS da API.

Regra	Origem	Destino
CNAME	<i>example.com</i>	d- <i>abcde12345</i> .execute-api. <i>us-east-2</i> .amazonaws.com

Etapa 4: Adicionar o domínio ao seu grupo de usuários do Amazon Cognito:

1. Acesse o [console do Amazon Cognito](#).
2. Selecione o link do grupo de usuários.
3. Escolha Integrações.
4. Em Domínio, escolha Ações, Criar domínio personalizado.
5. Insira seu domínio personalizado e selecione seu certificado ACM.
6. Escolha Criar domínio personalizado.

Etapa 5: Configurar o URL de retorno de chamada do API Gateway

1. Acesse o [console do Amazon Cognito](#).
2. Em sua integração de aplicativos do grupo de usuários do Amazon Cognito, Clientes e análises de aplicativos, selecione o link do aplicativo.
3. Na interface do usuário hospedada, escolha Editar.
4. Em URLs de retorno de chamada permitidos, escolha Adicionar outro URL e insira um URL de retorno de chamada, como `example.com/login`.

Etapa 6: Configurar uma função do Lambda

1. Navegue até o [console do Lambda](#).
2. No painel de navegação, escolha Funções.
3. Filtre a lista de funções para encontrar `ParallelClusterUIFunction` e selecione o link.
4. Escolha Configuração, Variáveis de ambiente.

5. Selecione a opção Editar.
6. Para o valor `SITE_URL`, insira seu domínio personalizado.
7. Navegue até seu domínio, (por exemplo, `example.com`) e autentique-se para se conectar à interface de usuário do AWS ParallelCluster.

Opções de login do grupo de usuários do Amazon Cognito

As seções a seguir se referem a links de CloudFormation criação rápida ou URLs de criação rápida. O URL de criação rápida leva você a um Assistente de criação de pilha, onde você pode fornecer entradas de criação rápida de modelo de pilha e pode implantar a pilha. Para obter mais informações sobre pilhas de CloudFormation criação rápida, consulte [Criação de links de criação rápida para pilhas](#) no Guia do usuário. AWS CloudFormation

Para manter um grupo de usuários do Amazon Cognito que você possa usar com várias instâncias de interface do usuário do AWS ParallelCluster, considere as seguintes opções:

- Use uma instância de AWS ParallelCluster interface de usuário existente vinculada a um grupo de usuários do Amazon Cognito criado a partir de uma pilha CloudFormation aninhada. Isso é o que será criado quando você implanta uma interface do usuário do AWS ParallelCluster usando o link de criação rápida e quando mantém todos os parâmetros do Amazon Cognito em branco.
- Use um grupo de usuários autônomo do Amazon Cognito que tenha sido implantado antes da implantação da interface do usuário do AWS ParallelCluster. Em seguida, implante uma nova instância de interface de usuário do AWS ParallelCluster vinculada ao grupo de usuários autônomo do Amazon Cognito que você já implantou. Dessa forma, você separa a implantação do Amazon Cognito da implantação da interface do usuário do AWS ParallelCluster. Além disso, CloudFormation pilhas de AWS ParallelCluster interface de usuário não aninhadas são mais fáceis de atualizar.

Use um grupo de usuários existente do Amazon Cognito com uma nova instância de interface de usuário do AWS ParallelCluster

1. No CloudFormation console, selecione a pilha de AWS ParallelCluster UI que contém o grupo de usuários do Amazon Cognito que você deseja usar com AWS ParallelCluster várias instâncias de UI.
2. Navegue até a pilha aninhada que criou o grupo de usuários do Amazon Cognito.
3. Selecione a guia Saídas.

4. Copie os valores dos parâmetros a seguir:
 - `UserPoolId`
 - `UserPoolAuthDomain`
 - `SNSRole`
5. Implante uma nova instância de interface do usuário do AWS ParallelCluster usando o link de criação rápida e preencha todos os parâmetros `External AWS ParallelCluster UI Amazon Cognito` com as saídas que você copiou. Isso impede que a nova pilha de interface do usuário AWS ParallelCluster crie um novo grupo e o vincula ao grupo de usuários existente do Amazon Cognito que foi criado a partir de uma pilha aninhada. Você pode implantar novas instâncias de interface de usuário do AWS ParallelCluster subsequentes que tenham os mesmos valores de parâmetros e vinculá-las ao grupo de usuários do Amazon Cognito.

Como criar um grupo de usuários autônomo do Amazon Cognito

Links de criação rápida do Amazon Cognito da interface de usuário do AWS ParallelCluster por região

Links de criação rápida do Amazon Cognito da interface de usuário

[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](#)

Links de criação rápida do Amazon Cognito da interface de usuário

[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](#)

1. Inicie uma pilha exclusiva do Amazon Cognito escolhendo um link de criação rápida rotulado com a mesma Região da AWS na qual você implanta suas instâncias de interface do usuário do AWS ParallelCluster. Consulte os links de criação rápida no início desta seção.
2. Após a conclusão da criação da pilha, selecione a guia Saídas e copie os valores dos seguintes parâmetros:
 - `UserPoolId`
 - `UserPoolAuthDomain`
 - `SNSRole`
3. Implante uma nova instância de interface do usuário do AWS ParallelCluster escolhendo um link de início rápido da interface do usuário do AWS ParallelCluster e preenchendo todos os parâmetros `External AWS ParallelCluster UI Amazon Cognito` com os valores que você copiou. A nova instância de interface de usuário do AWS ParallelCluster será vinculada ao grupo de usuários autônomo do Amazon Cognito e não criará uma pilha aninhada ou um novo grupo de usuários. Você pode implantar novas instâncias de interface de usuário do AWS ParallelCluster subsequentes que tenham os mesmos valores de parâmetros e vinculá-las ao grupo de usuários autônomo do Amazon Cognito.

Identifique a versão do AWS ParallelCluster e da interface do usuário do AWS ParallelCluster.

1. No CloudFormation console, selecione uma pilha de AWS ParallelCluster interface do usuário.
2. Selecione a guia Parâmetros.
3. A versão do AWS ParallelCluster é o valor do parâmetro Version.
4. A versão da AWS ParallelCluster interface do usuário está no final do PublicEcrImageUri valor. Por exemplo, se o valor for `public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02`, a versão será `2023.02`.

Atualizar a interface de usuário do AWS ParallelCluster para uma nova versão do AWS ParallelCluster

Para atualizar a interface do usuário do AWS ParallelCluster para a versão mais recente do AWS ParallelCluster, inicie uma nova pilha escolhendo um [link de criação rápida](#).

Custos de interface de usuário do AWS ParallelCluster

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria Free Tier AWS na maioria dos casos. A tabela a seguir lista os limites de Serviços da AWS que a interface do usuário do AWS ParallelCluster depende e seus limites de nível gratuito. Estima-se que o uso típico custe menos de um dólar por mês.

Serviço	Nível gratuito da AWS
Amazon Cognito	50.000 usuários ativos mensalmente
Amazon API Gateway	1 milhão de chamadas de API REST
AWS Lambda	1 milhão de solicitações gratuitas por mês e 400.000 GB-segundos de tempo de computação por mês
EC2 Image Builder	Sem custo, exceto EC2
Amazon Elastic Compute Cloud	Criação única de imagem de contêiner de 15 minutos

Serviço	Nível gratuito da AWS
AWS CloudFormation	5 GB de dados (ingestão, armazenamento de arquivos e dados verificados por consultas do Logs Insights)

Começando com AWS ParallelCluster

Comece configurando e criando um cluster usando a interface de linha de AWS ParallelCluster comando (CLI) ou a interface de usuário (UI) baseada na web. A AWS ParallelCluster interface do usuário foi adicionada na versão 3.5.0.

Tópicos

- [Configure e crie um cluster com a interface de linha de AWS ParallelCluster comando](#)
- [Configurar e criar um cluster com a interface do usuário do AWS ParallelCluster](#)
- [Conectar-se a um cluster](#)

Configure e crie um cluster com a interface de linha de AWS ParallelCluster comando

Depois de instalar AWS ParallelCluster, conclua as etapas de configuração a seguir.

Verifique se sua AWS conta tem uma função que inclui as permissões necessárias para executar a `pcluster` CLI. Para ter mais informações, consulte [AWS ParallelCluster exemplo de políticas pcluster de usuário](#).

Configure suas AWS credenciais. Para obter mais informações, consulte [Configurar a AWS CLI](#) no Guia do usuário da 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]:
```

O Região da AWS local em que o cluster é lançado deve ter pelo menos um par de chaves do Amazon EC2. Para obter mais informações, consulte [Pares de chaves do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Ao usar a interface de linha de AWS ParallelCluster comando (CLI), você paga somente pelos AWS recursos criados ao criar ou atualizar AWS ParallelCluster imagens e clusters. Para ter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

Criar e configurar seu primeiro cluster

Crie seu primeiro cluster usando o comando CLI `pcluster configure` para iniciar um assistente que solicita todas as informações necessárias para configurar e criar seu cluster. Os detalhes da sequência diferem quando usada AWS Batch como programador em comparação com a utilizaçãoSlurm.

Slurm

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

Na lista de Região da AWS identificadores válidos, escolha Região da AWS onde você deseja que seu cluster seja executado.

Note

A lista Regiões da AWS exibida é baseada na partição da sua conta e inclui apenas as Regiões da AWS que estão habilitadas para sua conta. Para obter mais informações sobre como habilitar Regiões da AWS sua conta, consulte [Gerenciando Regiões da AWS](#) no Referência geral da AWS. O exemplo mostrado é da partição AWS global. Se sua conta estiver na AWS GovCloud (US) partição, somente Regiões da AWS nessa partição serão listadas (`gov-us-east-1` e `gov-us-west-1`). Da mesma forma, se sua conta estiver na partição AWS da China, somente `cn-north-1` e `cn-northwest-1` serão mostradas. Para obter a lista completa dos Regiões da AWS produtos suportados por AWS ParallelCluster, consulte [Regiões compatíveis do AWS ParallelCluster](#).

Allowed values for Região da 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. ca-central-1
9. eu-central-1
10. eu-north-1
11. eu-south-1
12. eu-west-1
13. eu-west-2
14. eu-west-3
15. me-south-1
16. sa-east-1
17. us-east-1
18. us-east-2
19. us-west-1
20. us-west-2

Região da AWS ID [ap-northeast-1]:

O par de chaves é selecionado a partir dos pares de chaves registrados com o Amazon EC2 na Região da AWS selecionada. Escolha o par de chaves:

Allowed values for EC2 Key Pair Name:

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

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

Escolha o programador a ser usado com seu cluster.

Allowed values for Scheduler:

1. slurm
2. awsbatch

Scheduler [slurm]:

Escolha o sistema operacional.

Allowed values for Operating System:

1. alinux2
2. centos7
3. ubuntu2204
4. ubuntu2004

```
5. rhel8
Operating System [alinux2]:
```

Escolha o tipo de instância do nó principal:

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

Escolha a configuração da fila. Observação: o tipo de instância não pode ser especificado para vários recursos de computação na mesma fila.

```
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]:
```

Permita que o EFA execute aplicativos que exigem altos níveis de comunicação entre instâncias em grande escala, sem AWS custo adicional:


- Escolha um tipo de instância que seja [compatível com Elastic Fabric Adapter \(EFA\)](#).
- Ative o [EFA](#).
- Especifique um nome para um [Placement Group](#) existente. Se você deixar em branco, AWS ParallelCluster cria um para você.

```
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 []:
```

Depois que as etapas anteriores forem concluídas, decida se quer usar uma VPC existente ou AWS ParallelCluster criar uma VPC para você. Se você não tiver uma VPC configurada corretamente, AWS ParallelCluster poderá criar uma nova para você. Ela coloca os nós principais e de computação na mesma sub-rede pública ou somente o nó principal em uma sub-rede pública com todos os nós de computação em uma sub-rede privada. Se você permitir a AWS ParallelCluster criação de uma VPC, deverá decidir se todos os nós devem estar em uma sub-rede pública. Para ter mais informações, consulte [Configurações de rede](#).


Se você configurar seu cluster para usar tipos de instância que tenham várias interfaces de rede ou uma placa de rede, consulte [Configurações de rede](#) para obter mais requisitos de rede.

É possível atingir sua cota para o número de VPCs permitido em uma Região da AWS. A cota padrão é de cinco VPCs para uma Região da AWS. Para obter mais informações sobre essa cota e como solicitar um aumento, consulte [VPC e sub-redes](#) no Guia de usuário do Amazon VPC.

 Important

As VPCs criadas por AWS ParallelCluster não habilitam os registros de fluxo de VPC por padrão. Os logs de fluxo da VPC permitem capturar informações sobre tráfego IP de entrada e de saída nas interfaces de rede de seus VPCs. Para obter mais informações, consulte [Logs de fluxo da VPC](#) no Guia do usuário do Amazon Virtual Private Cloud.

Se você deixar AWS ParallelCluster criar uma VPC, certifique-se de decidir se todos os nós devem estar em uma sub-rede pública.

 Note

Se você escolher 1. Head node in a public subnet and compute fleet in a private subnet, o AWS ParallelCluster cria um gateway NAT que resulta em custo adicional, mesmo se você especificar recursos de nível 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 você não criar uma nova VPC, você deverá selecionar uma VPC existente.

Se você optar por AWS ParallelCluster criar a VPC, anote a ID da VPC para poder usá-la para excluí-la posteriormente AWS CLI .

```
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
```

Depois de selecionar a VPC, decida se deseja usar sub-redes existentes ou criar novas.

```
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 config-file.yaml
```

Na lista de Região da AWS identificadores válidos, escolha Região da AWS onde você deseja que seu cluster seja executado.

Note

A lista Regiões da AWS exibida é baseada na partição da sua conta. Inclui apenas os Regiões da AWS que estão habilitados para sua conta. Para obter mais informações sobre como habilitar Regiões da AWS sua conta, consulte [Gerenciando Regiões da AWS](#) no Referência geral da AWS. O exemplo mostrado é da partição AWS global. Se sua conta estiver na AWS GovCloud (US) partição, somente Regiões da AWS nessa partição serão listadas (gov-us-east-1gov-us-west-1). Da mesma forma, se sua conta

estiver na partição AWS da China, somente `cn-north-1` e `cn-northwest-1` serão mostradas. Para obter a lista completa dos Regiões da AWS produtos suportados por AWS ParallelCluster, consulte [Regiões compatíveis do AWS ParallelCluster](#).

Allowed values for Região da 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. `ca-central-1`
9. `eu-central-1`
10. `eu-north-1`
11. `eu-south-1`
12. `eu-west-1`
13. `eu-west-2`
14. `eu-west-3`
15. `me-south-1`
16. `sa-east-1`
17. `us-east-1`
18. `us-east-2`
19. `us-west-1`
20. `us-west-2`

Região da AWS ID [`us-east-1`]:

O par de chaves é selecionado a partir dos pares de chaves registrados com o Amazon EC2 na Região da AWS selecionada. Escolha o par de chaves:

Allowed values for EC2 Key Pair Name:

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

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

Escolha o programador a ser usado com seu cluster.

Allowed values for Scheduler:

1. `slurm`
2. `awsbatch`

```
Scheduler [slurm]: 2
```

Quando `awsbatch` é selecionado como o programador, `alinux2` é usado como o sistema operacional. O tipo de instância do nó principal é inserido.

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

Escolha a configuração da fila. O AWS Batch agendador contém apenas uma única fila. O tamanho máximo do cluster de nós de computação é inserido. Isso é medido em vCPUs.

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

Decida se quer usar VPCs existentes ou deixar AWS ParallelCluster criar VPCs para você. Se você não tiver uma VPC configurada corretamente, o AWS ParallelCluster poderá criar uma nova. Ela usa os nós principais e de computação na mesma sub-rede pública ou somente o nó principal em uma sub-rede pública com todos os nós em uma sub-rede privada. É possível atingir sua cota para o número de VPCs permitido em uma Região. O número padrão de VPCs é cinco. Para obter mais informações sobre essa cota e como solicitar um aumento, consulte [VPC e sub-redes](#) no Guia de usuário do Amazon VPC.

Important

As VPCs criadas por AWS ParallelCluster não habilitam os registros de fluxo de VPC por padrão. Os logs de fluxo da VPC permitem capturar informações sobre tráfego IP de entrada e de saída nas interfaces de rede de seus VPCs. Para obter mais informações, consulte [Logs de fluxo da VPC](#) no Guia do usuário do Amazon Virtual Private Cloud.

Se você deixar AWS ParallelCluster criar uma VPC, certifique-se de decidir se todos os nós devem estar em uma sub-rede pública.

Note

Se você escolher `1. Head node in a public subnet and compute fleet in a private subnet`, o AWS ParallelCluster cria um gateway NAT que resulta em custo adicional, mesmo se você especificar recursos de nível 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 você não criar uma nova VPC, você deverá selecionar uma VPC existente.

Se você optar por AWS ParallelCluster criar a VPC, anote a ID da VPC para poder usá-la AWS CLI ou AWS Management Console excluí-la posteriormente.

```

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

```

Depois de selecionar a VPC, confirme a decisão entre usar sub-redes existentes ou criar novas.

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

```

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

```

Depois de concluir as etapas anteriores, um cluster simples é iniciado em uma VPC. A VPC usa uma sub-rede existente que oferece suporte a endereços IP públicos. A tabela de rotas para a sub-rede é `0.0.0.0/0 => igw-xxxxxx`. Atenção às condições a seguir:

- A VPC deve ter `DNS Resolution = yes` e `DNS Hostnames = yes`.
- A VPC também deve ter opções DHCP com o `domain-name` correto para a Região da AWS. O conjunto de opções DHCP padrão já especifica o `AmazonProvidedDNS` necessário. Se estiver especificando mais de um servidor de nomes de domínio, consulte [Conjuntos de opções DHCP](#) no Amazon VPC User Guide. Ao usar sub-redes privadas, use um gateway NAT ou um proxy interno para permitir o acesso à web para nós de computação. Para ter mais informações, consulte [Configurações de rede](#).

Quando todas as configurações tiverem valores válidos, você poderá inicializar o cluster executando o comando de criação.

```
$ 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": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  },
  "validationMessages": []
}
```

Acompanhe o progresso do cluster:

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

ou

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

Quando o cluster atingir o status `"clusterStatus": "CREATE_COMPLETE"`, você poderá se conectar a ele usando as configurações normais de cliente SSH. Para obter mais informações sobre como conectar com instâncias do Amazon EC2, consulte [Guia do usuário do EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux. Ou você pode conectar o cluster por meio de

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

Para excluir o cluster, execute o comando a seguir.

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

Depois que o cluster for excluído, você poderá excluir os recursos de rede na VPC excluindo a CloudFormation pilha de rede. O nome da pilha começa com "parallelclusternetworking-" e contém a hora de criação no formato "YYYYMMDDHHMMSS". Você pode listar as pilhas usando o comando [list-stacks](#).

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

A pilha pode ser excluída usando o comando [delete-stack](#).

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

A VPC que [pcluster configure](#) cria para você não é criada na pilha de CloudFormation rede. Você pode excluir essa VPC manualmente no console ou usando a AWS CLI.

```
$ aws --region us-east-1 ec2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

Configurar e criar um cluster com a interface do usuário do AWS ParallelCluster

A AWS ParallelCluster interface do usuário é uma interface de usuário baseada na Web que espelha a AWS ParallelCluster `pcluster` CLI, ao mesmo tempo em que fornece uma experiência semelhante à de um console. Você instala e acessa a AWS ParallelCluster interface do usuário no seu Conta da AWS. Quando você a executa, a AWS ParallelCluster interface do usuário acessa uma instância da AWS ParallelCluster API hospedada no Amazon API Gateway em seu Conta da AWS.

Note

O assistente de AWS ParallelCluster interface do usuário pode não ter opções de interface do usuário para todos os recursos compatíveis na AWS ParallelCluster versão mais recente compatível. Você pode editar manualmente o arquivo de configuração conforme necessário ou usar a AWS ParallelCluster CLI.

Nesta seção, orientaremos você na configuração e criação de um cluster usando a interface do usuário do AWS ParallelCluster .

Pré-requisitos:

- Acesso a uma instância de AWS ParallelCluster UI em execução. Para ter mais informações, consulte [Instalando a interface do usuário do AWS ParallelCluster](#).

Configurar e criar um cluster

1. Na visualização AWS ParallelCluster UI Clusters, escolha Create cluster, Step by step.
2. Em Cluster, Nome, insira um nome para o seu cluster.
3. Escolha uma VPC com uma sub-rede pública para seu cluster e escolha Avançar.
4. No nó principal, escolha Adicionar sessão SSM e escolha Avançar.
5. Em Filas, Recursos de computação, escolha 1 para Nós estáticos.
6. Em Tipo de instância, remova o tipo de instância padrão selecionado, escolha t2.micro e escolha Avançar.
7. Em Armazenamento, escolha Avançar.
8. Em Configuração do cluster, revise a configuração do cluster YAML e escolha Dry run para validá-la.
9. Escolha Criar para criar seu cluster, com base na configuração validada.
10. Depois de alguns segundos, a AWS ParallelCluster interface do usuário leva você automaticamente de volta aos Clusters, onde você pode monitorar o status de criação do cluster e os eventos de pilha.
11. Escolha Detalhes para ver os detalhes do cluster, como a versão e o status.
12. Escolha Instâncias para ver a lista de instâncias e status do EC2.

13. Escolha Stack events para visualizar os eventos da pilha do cluster e um AWS Management Console link para a CloudFormation pilha que cria o cluster.
14. Em Detalhes, após a conclusão da criação do cluster, escolha Exibir YAML para visualizar ou baixar o arquivo YAML de configuração do cluster.
15. Após a conclusão da criação do cluster, escolha Shell para acessar o nó principal do cluster.

Note

Quando você escolhe o Shell, AWS ParallelCluster abre uma sessão do Amazon EC2 Systems Manager e adiciona `ssm-user` um a `/etc/sudoers` Para obter mais informações, consulte [Ativar ou desativar as permissões administrativas da conta do `ssm-user`](#) no Guia do usuário do Amazon EC2 Systems Manager.

16. Para limpar, na visualização Clusters, selecione o cluster e escolha Ações, Excluir cluster.

Conectar-se a um cluster

Ao usar o AWS ParallelCluster, você pode se conectar ao nó principal do cluster para executar trabalhos, visualizar resultados, gerenciar usuários e monitorar o cluster e o status do trabalho. Conecte-se à instância do nó principal do cluster usando os seguintes métodos:

- Faça login usando `ssh` com uma [key pair](#). Especifique a chave privada em [HeadNode](#) / [KeyName](#) na configuração do cluster. Para obter mais informações, consulte [Conectar-se à instância do Linux usando SSH](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.
- Faça o login usando o comando `pcluster ssh` da interface de linha de comandos (CLI). Especifique a chave privada em [HeadNode](#) / [KeyName](#) na configuração do cluster. Para obter mais informações, consulte [pcluster ssh](#).
- Conecte-se ao nó principal do cluster usando uma sessão SSM. Você deve adicionar a política gerenciada `AmazonSSMManagedInstanceCore` ao [HeadNode](#) / [AdditionalIamPolicies](#) na configuração do cluster para se conectar usando uma sessão SSM. Para obter mais informações, consulte [Gerenciador de sessões SSM](#) no Guia do usuário do SSM.
- Conecte-se ao nó principal do cluster usando o NICE DCV. Para obter mais informações, consulte [Conecte-se ao nó principal por meio do NICE DCV](#).
- Ao usar a interface de usuário do AWS ParallelCluster, você também pode se conectar ao nó principal do cluster usando um comando EC2 Connect fornecido pela interface do usuário.

Acesso de vários usuários aos clusters

Aprenda a implementar e gerenciar o acesso de vários usuários a um único cluster.

Neste tópico, um usuário do AWS ParallelCluster se refere a um usuário do sistema para instâncias de computação. Um exemplo é um `ec2-user` para uma instância do EC2 da AWS.

O suporte de acesso multiusuário do AWS ParallelCluster está disponível em todas as Regiões da AWS onde o AWS ParallelCluster está disponível atualmente. Ele funciona com outros Serviços da AWS, incluindo o [Amazon FSx for Lustre](#) e o [Amazon Elastic File System](#).

Você pode usar um [AWS Directory Service for Microsoft Active Directory](#) ou [Simple AD](#) para gerenciar o acesso ao cluster. Certifique-se de verificar a [disponibilidade na Região da AWS](#) para esses serviços. Para configurar um cluster, especifique uma [AWS ParallelCluster DirectoryService](#) configuração. AWS Directory Service diretórios podem ser conectados a vários clusters. Isso permite o gerenciamento centralizado de identidades em vários ambientes e uma experiência de login unificada.

Ao usar AWS Directory Service para acesso de vários usuários do AWS ParallelCluster, você pode fazer login no cluster com as credenciais do usuário definidas no diretório. Essas credenciais consistem em um nome e senha. Depois de fazer login no cluster pela primeira vez, uma chave SSH do usuário é gerada automaticamente. Você pode usá-la para fazer login sem uma senha.

Você pode criar, excluir e modificar os usuários ou grupos de um cluster após a implantação do serviço de diretório. Com AWS Directory Service, você pode fazer isso no AWS Management Console ou usando a ferramenta Usuários e Computadores do Active Directory. Essa ferramenta pode ser acessada por qualquer instância do EC2 associada ao seu Active Directory. Para obter mais informações, consulte [Instalação das ferramentas de administração do Active Directory](#).

Se você planeja usar o AWS ParallelCluster em uma única sub-rede sem acesso à Internet, consulte [AWS ParallelCluster em uma sub-rede única sem acesso à internet](#) para ver os requisitos adicionais.

Tópicos

- [Crie um Active Directory](#)
- [Crie um cluster com um domínio AD](#)
- [Faça login em um cluster integrado a um domínio do AD](#)
- [Executando um trabalho de MPI](#)
- [Exemplo de AWS Managed Microsoft AD sobre configurações de cluster LDAP\(S\)](#)

Crie um Active Directory

Certifique-se de criar um Active Directory (AD) antes de criar seu cluster. Para obter informações sobre como escolher o tipo de diretório ativo para seu cluster, consulte [Qual escolher](#) no Guia de Administração do AWS Directory Service.

Se o diretório estiver vazio, adicione usuários com nomes de usuário e senhas. Para obter mais informações, consulte a documentação específica do [AWS Directory Service for Microsoft Active Directory](#) ou do [Simple AD](#).

Note

O AWS ParallelCluster exige que todos os diretórios de usuários do Active Directory estejam no diretório `/home/$user`.

Crie um cluster com um domínio AD

Warning

Esta seção introdutória descreve como configurar AWS ParallelCluster um servidor Gerenciado do Active Directory (AD) por meio do Lightweight Directory Access Protocol (LDAP). O LDAP é um protocolo inseguro. Para sistemas de produção, é altamente recomendável o uso de certificados TLS (LDAPS), conforme descrito na seção [Exemplo de AWS Managed Microsoft AD sobre configurações de cluster LDAP\(S\)](#) a seguir.

Configure seu cluster para que ele se integre a um diretório especificando as informações relevantes na seção `DirectoryService` do arquivo de configuração do cluster. Para obter mais informações, consulte a seção configuração [DirectoryService](#).

Você pode usar o exemplo a seguir para integrar seu cluster com um AWS Managed Microsoft AD sobre o protocolo Lightweight Directory Access Protocol (LDAP).

Definições específicas que são necessárias para uma configuração AWS Managed Microsoft AD sobre o LDAP:

- Você deve definir o parâmetro `ldap_auth_disable_tls_never_use_in_production` como `True` via [DirectoryService](#) / [AdditionalSssdConfigs](#).

- Você pode especificar nomes de host do controlador ou endereços IP para [DirectoryService](#) / [DomainAddr](#).
- A sintaxe [DirectoryService](#) / [DomainReadOnlyUser](#) deve ser conforme segue:

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

Obtenha seus dados de configuração AWS Managed Microsoft AD:

```
$ 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"
        ]
      }
    }
  ]
}
```

Configuração de cluster para 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
```

Para usar essa configuração para um Simple AD, altere o valor da propriedade

DomainReadOnlyUser na seção **DirectoryService**:

```
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
```

Considerações:

- Recomendamos que você use LDAP sobre TLS/SSL (ou LDAPS) em vez de apenas LDAP. O TLS/SSL garante que a conexão seja criptografada.

- O valor da propriedade [DirectoryService / DomainAddr](#) corresponde às entradas na lista DnsIpAddrs da saída `describe-directories`.
- Recomendamos que seu cluster use sub-redes localizadas na mesma zona de disponibilidade para a qual [DirectoryService / DomainAddr](#) aponta. Se você usa uma [configuração personalizada do Protocolo de Configuração Dinâmica de Host \(DHCP\)](#) recomendada para VPCs de diretórios e suas sub-redes não estão localizadas na Zona de Disponibilidade [DirectoryService / DomainAddr](#), é possível realizar o tráfego cruzado entre as Zonas de Disponibilidade. O uso de configurações DHCP personalizadas não é necessário para usar o recurso de integração multiusuário do AD.
- O valor da propriedade [DirectoryService / DomainReadOnlyUser](#) especifica um usuário que deve ser criado no diretório. Esse usuário não é criado por padrão. Recomendamos que você não dê permissão a esse usuário para modificar os dados do diretório.
- O valor de propriedade [DirectoryService / PasswordSecretArn](#) aponta para um segredo AWS Secrets Manager que contém a senha do usuário que você especificou para a propriedade [DirectoryService / DomainReadOnlyUser](#). Se a senha desse usuário mudar, atualize o valor secreto e atualize o cluster. Para atualizar o cluster para o novo valor secreto, você deve interromper a frota de computação com o comando `pcluster update-compute-fleet`. Se você configurou seu cluster para usar [LoginNodes](#), interrompa o [LoginNodes / Pools](#) e atualize o cluster depois de definir o [LoginNodes / Pools / Count](#) para 0. Em seguida, execute o comando a seguir a partir do nó principal do cluster.

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

Para obter outro exemplo, consulte também [Integração do Active Directory](#).

Faça login em um cluster integrado a um domínio do AD

Se você ativou o recurso de integração de domínio Active Directory (AD), a autenticação por senha será habilitada no nó principal do cluster. O diretório inicial de um usuário do AD é criado no primeiro login do usuário no nó principal ou na primeira vez que um `sudo-user` troca para o usuário do AD no nó principal.

A autenticação por senha não fica habilitada para nós de computação do cluster. Os usuários do AD devem fazer login nos nós de computação com chaves SSH.

Por padrão, as chaves SSH são configuradas no diretório `/${HOME}/.ssh` de usuários do AD no primeiro login SSH no nó principal. Esse comportamento pode ser desabilitado definindo a propriedade boolean [DirectoryService / GenerateSshKeysForUsers](#) como `false` na configuração do cluster. Por padrão, [DirectoryService / GenerateSshKeysForUsers](#) é definido como `true`.

Se uma aplicação do AWS ParallelCluster exigir SSH sem senha entre os nós do cluster, verifique se as chaves SSH estão configuradas corretamente no diretório inicial do usuário.

As senhas AWS Managed Microsoft AD expiram em 42 dias. Para obter mais informações, consulte [Gerenciar políticas de senhas para AWS Managed Microsoft AD](#) no Guia de administração do AWS Directory Service. Se sua senha expirar, ela deverá ser redefinida para restaurar o acesso ao cluster. Para obter mais informações, consulte [Como redefinir uma senha de usuário e senhas expiradas](#).

Note

Se o recurso de integração do AD não funcionar conforme o esperado, os logs do SSSD podem fornecer informações úteis de diagnóstico para solucionar o problema. Esses logs estão localizados no diretório `/var/log/sss` nos nós do cluster. Por padrão, eles também são armazenados no grupo de CloudWatch logs da Amazon de um cluster.

Para obter mais informações, consulte [Solução de problemas de integração de vários usuários com o Active Directory](#).

Executando um trabalho de MPI

Conforme sugerido no SchedMD, inicie trabalhos MPI usando Slurm como método de bootstrap do MPI. Para obter mais informações, consulte a [documentação oficial do Slurm](#) ou a documentação oficial da sua biblioteca MPI.

Por exemplo, na [documentação oficial do IntelMPI](#), você aprende que, ao executar um trabalho StarCCM, você deve definir Slurm como orquestrador de processos exportando a variável de ambiente `I_MPI_HYDRA_BOOTSTRAP=slurm`.

Note

Problema conhecido

No caso em que seu aplicativo MPI depende do SSH como mecanismo para gerar trabalhos de MPI, é possível que resulte em um [bug conhecido no Slurm](#) que causa a resolução errada do nome de usuário do diretório para "ninguém".

Configure seu aplicativo para usar Slurm como método de bootstrap do MPI ou consulte [Problemas conhecidos com a resolução do nome de usuário](#) na seção de Solução de problemas para obter mais detalhes e possíveis soluções alternativas.

Exemplo de AWS Managed Microsoft AD sobre configurações de cluster LDAP(S)

O AWS ParallelCluster oferece suporte ao acesso de vários usuários por meio da integração com AWS Directory Service sobre o Lightweight Directory Access Protocol (LDAP) ou com o LDAP sobre TLS/SSL (LDAPS).

Os exemplos a seguir mostram como criar configurações de cluster para integração com um AWS Managed Microsoft AD sobre LDAP(S).

AWS Managed Microsoft AD sobre LDAPS com verificação de certificado

Você pode usar este exemplo para integrar seu cluster com um AWS Managed Microsoft AD sobre LDAPS, com verificação de certificado.

Definições específicas para uma configuração AWS Managed Microsoft AD sobre LDAPS com configuração de certificados:

- [DirectoryService](#) / [LdapTlsReqCert](#) deve ser definido como `hard` (padrão) para LDAPS com verificação de certificado.
- [DirectoryService](#) / [LdapTlsCaCert](#) deve especificar o caminho para seu certificado de autoridade (CA).

O certificado CA é um pacote de certificados que contém os certificados de toda a cadeia de CA que emitiu certificados para os controladores de domínio AD.

Seu certificado CA e seus certificados devem estar instalados nos nós do cluster.

- Os nomes de host dos controladores devem ser especificados para [DirectoryService](#) / [DomainAddr](#), não para endereços IP.
- A sintaxe [DirectoryService](#) / [DomainReadOnlyUser](#) deve ser conforme segue:

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

Exemplo de arquivo de configuração de cluster para usar o AD sobre 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://aws-parallelcluster/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://aws-parallelcluster-pcluster/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/openldap/cacerts/corp.example.com.bundleca.cer
LdapTlsReqCert: hard
```

Adicione certificados e configure controladores de domínio no script de pós-instalação:

```
*#!/bin/bash*
set -e

AD_CERTIFICATE_S3_URI="s3://corp.example.com/bundle/corp.example.com.bundleca.cer"
AD_CERTIFICATE_LOCAL="/etc/openldap/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
```

Você pode recuperar os nomes de host dos controladores de domínio das instâncias associadas ao domínio, conforme mostrado nos exemplos a seguir.

Da instância do 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
```

Da instância do 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 sobre LDAPS sem verificação de certificado

Você pode usar este exemplo para integrar seu cluster com um AWS Managed Microsoft AD sobre LDAPS, com verificação de certificado.

Definições específicas para uma AWS Managed Microsoft AD sobre LDAPS sem uma configuração de verificação de certificados:

- [DirectoryService](#) / [LdapTlsReqCert](#) deve ser definido como `never`.
- Você pode especificar nomes de host do controlador ou endereços IP para [DirectoryService](#) / [DomainAddr](#).
- A sintaxe [DirectoryService](#) / [DomainReadOnlyUser](#) deve ser conforme segue:

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

Exemplo de arquivo de configuração de cluster para usar AWS Managed Microsoft AD sobre LDAPS sem verificação de certificado:

```
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
```

Práticas recomendadas

Práticas recomendadas: seleção do tipo de instância do nó principal

Mesmo que o nó principal não execute uma tarefa, suas funções e seu tamanho são cruciais para o desempenho geral do cluster. Ao escolher o tipo de instância a ser usado em seu nó principal, considere as seguintes características:

Tamanho do cluster: O nó principal orquestra a lógica de escalonamento do cluster e é responsável por anexar novos nós ao programador. Para aumentar e reduzir a escala de um cluster que tem um grande número de nós, forneça um pouco mais de capacidade computacional ao nó principal.

Sistemas de arquivos compartilhados: ao usar sistemas de arquivos compartilhados, escolha um tipo de instância com largura de banda da rede e largura de banda do Amazon EBS suficientes para lidar com seus fluxos de trabalho. Certifique-se de que o nó principal seja capaz de expor diretórios de servidor NFS suficientes para o cluster e lidar com os artefatos que precisam ser compartilhados entre os nós de computação e o nó principal.

Práticas recomendadas: desempenho da rede

O desempenho da rede é fundamental para aplicações de computação de alta performance (HPC). Sem um desempenho de rede confiável, esses aplicativos não funcionam conforme o esperado. Para otimizar o desempenho da rede, considere as práticas recomendadas a seguir.

- Grupo de posicionamento: se você estiver usando o Slurm, considere configurar cada fila do Slurm para usar um grupo de posicionamento de cluster. Um grupo de posicionamento de cluster é um agrupamento lógico de instâncias dentro de uma única zona de disponibilidade. Para obter mais informações, consulte [grupos de posicionamento](#) no Guia do usuário do Amazon EC2 para instâncias Linux. Você pode especificar um [PlacementGroup](#) na seção [Networking](#) da fila, cada recurso de computação é atribuído ao grupo de posicionamento da fila. Ao especificar um [PlacementGroup](#) na seção [Networking](#) do recurso de computação, esse recurso de computação específico é atribuído ao grupo de posicionamento. A especificação do grupo de posicionamento de recursos de computação substitui a especificação da fila para o recurso de computação. Para obter mais informações, consulte [SlurmQueues / Networking / PlacementGroup](#) e [SlurmQueues / ComputeResources / Networking / PlacementGroup](#).

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

Como alternativa, use AWS ParallelCluster para criar um grupo de posicionamento para você.

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

A partir do AWS ParallelCluster versão 3.3.0, a criação e o gerenciamento do grupo de posicionamento foram modificados. Quando você especifica o grupo de posicionamento a ser ativado, sem name ou Id, na fila, cada recurso de computação recebe seu próprio grupo de posicionamento gerenciado, em vez de um grupo gerenciado para toda a fila. Isso ajuda a reduzir erros de capacidade insuficiente. Se você precisar ter um grupo de posicionamento para toda a fila, poderá usar um grupo de posicionamento nomeado.

[SlurmQueues / Networking / PlacementGroup / Name](#) foi adicionado como alternativa preferencial para [SlurmQueues / Networking / PlacementGroup / Id](#).

Para obter mais informações, consulte [Networking](#).

- Rede aprimorada: considere escolher um tipo de instância que ofereça suporte à rede avançada. Essa recomendação se aplica a todas as [instâncias da geração atual](#). Para obter mais informações, consulte [rede aprimorada no Linux](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

- **Elastic Fabric Adapter:** para oferecer suporte a altos níveis de comunicação escalável entre instâncias, considere escolher interfaces de rede EFA para sua rede. O hardware de desvio de sistema operacional (SO) personalizado do EFA aprimora as comunicações entre instâncias com a elasticidade e flexibilidade sob demanda da Nuvem AWS. Você pode configurar cada [ComputeResource](#) da fila do Slurm para usar [Efa](#). Para obter mais informações sobre como usar EFA com o AWS ParallelCluster, consulte [Elastic Fabric Adapter](#).

```
ComputeResources:
```

```
- Name: your-compute-resource-name
```

```
  Efa:
```

```
    Enabled: true
```

Para obter mais informações sobre o EFA, consulte [Elastic Fabric Adapter](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

- **Largura de banda da instância:** a largura de banda é dimensionada com o tamanho da instância. Para obter informações sobre os diferentes tipos de instância, consulte [Instâncias otimizadas para o Amazon EBS](#) e [tipos de volume do Amazon EBS](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Práticas recomendadas: alertas de orçamento

Para gerenciar os custos dos recursos no AWS ParallelCluster, recomendamos que você use ações AWS Budgets para criar um orçamento. Você também pode criar alertas de limite de orçamento definidos para recursos selecionados da AWS. Para obter mais informações, consulte [Como configurar uma ação de orçamento](#) no Manual do usuário AWS Budgets. Da mesma forma, você também pode usar o Amazon CloudWatch para criar um alarme de cobrança. Para mais informações, consulte [Criar um alarme de faturamento para monitorar suas cobranças estimadas da AWS](#).

Práticas recomendadas: mover um cluster para uma nova versão de patch ou secundária do AWS ParallelCluster.

Atualmente, cada versão secundária do AWS ParallelCluster é independente e acompanha sua própria CLI `pccluster`. Para mover um cluster para uma nova versão secundária ou de patch, você deve recriar o cluster usando a CLI da nova versão.

Para otimizar o processo de mover um cluster para uma nova versão secundária ou de patch, recomendamos fazer o seguinte:

- Salve dados pessoais em volumes externos criados fora do cluster, como o Amazon EFS e o FSx for Lustre. Ao fazer isso, você pode mover facilmente os dados de um cluster para outro no futuro.
- Crie sistemas de armazenamento compartilhado usando os seguintes tipos. Você pode criar esses sistemas usando o AWS CLI ou o AWS Management Console.
 - [SharedStorage](#) / [EbsSettings](#) / [VolumeId](#)
 - [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)
 - [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)

Defina um sistema de arquivos ou volume em uma configuração de cluster como sistema de arquivos ou volume existente. Dessa forma, eles serão preservados quando você excluir o cluster e podem ser anexados a um novo cluster.

Recomendamos que você use o Amazon EFS ou sistemas de arquivos FSx for Lustre. Esses dois sistemas podem ser anexados a vários clusters ao mesmo tempo. Além disso, você pode anexar qualquer um desses sistemas a um novo cluster antes de excluir o cluster existente.

- Use [ações de bootstrap personalizadas](#) para personalizar suas instâncias em vez de usar uma AMI personalizada. Se, em vez disso, você usar uma AMI personalizada, precisará excluir e recriar essa AMI para cada nova versão lançada.
- Recomendamos aplicar as recomendações anteriores na seguinte sequência:
 1. Atualize a configuração do cluster existente para usar as definições existentes do sistema de arquivos.
 2. Verifique a versão do `pcluster` e atualize-a, se necessário.
 3. Crie e teste o novo cluster. Ao testar o novo cluster, verifique o seguinte:
 - Verifique se os dados estão disponíveis no cluster novo.
 - Verifique se a aplicação funciona no cluster novo.
 4. Depois que seu novo cluster estiver totalmente testado e operacional e você não precisar mais do cluster existente, exclua-o.

Passando do AWS ParallelCluster 2.x para 3.x

Ações de bootstrap personalizadas

Com o AWS ParallelCluster 3, você pode especificar diferentes scripts de ações de bootstrap personalizados para o nó principal e os nós de computação usando parâmetros `OnNodeStart` (`pre_install` no AWS ParallelCluster versão 2) e `OnNodeConfigured` (`post_install` no AWS ParallelCluster versão 2) nas seções [HeadNode](#) e [Scheduling / SlurmQueues](#). Para obter mais informações, consulte [Ações de bootstrap personalizadas](#).

Os scripts de ações de bootstrap personalizados desenvolvidos para o AWS ParallelCluster 2 devem ser adaptados para serem usados no AWS ParallelCluster 3:

- Não recomendamos usar `/etc/parallelcluster/cfnconfig` e `cfn_node_type` para diferenciar entre nós principais e de computação. Em vez disso, recomendamos que você especifique dois scripts diferentes em [HeadNode](#) e [Scheduling / SlurmQueues](#).
- Se você preferir continuar carregando `/etc/parallelcluster/cfnconfig` para uso em seu script de ações de bootstrap, observe que o valor `cfn_node_type` de foi alterado de "MasterServer" para "HeadNode" (consulte: [Linguagem inclusiva](#)).
- No AWS ParallelCluster 2, o primeiro argumento de entrada para scripts de ação de bootstrap foi o URL do S3 para o script e era reservado. No AWS ParallelCluster 3, somente os argumentos definidos na configuração são passados para os scripts.

Warning

O uso de variáveis internas fornecidas pelo arquivo `/etc/parallelcluster/cfnconfig` não é oficialmente compatível. Esse arquivo pode ser removido como parte de uma versão futura.

AWS ParallelCluster 2.x e 3.x usam sintaxe de arquivo de configuração diferente

A configuração do AWS ParallelCluster 3.x usa a sintaxe YAML. A referência completa pode ser encontrada em [Arquivos de configuração](#).

Além de exigir um formato de arquivo YAML, várias seções de configuração, definições de atributos e valores de parâmetros foram atualizados na versão AWS ParallelCluster 3.x. Nesta seção, observamos as principais mudanças na configuração do AWS ParallelCluster, juntamente com exemplos lado a lado que ilustram essas diferenças em cada versão do AWS ParallelCluster.

Exemplo de configuração de várias filas do programador com hyperthreading ativado e desativado

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

```

Exemplo da nova configuração do sistema de arquivos FSx for 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://bucket
export_path = s3://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://bucket
    ExportPath: s3://bucket/export_dir
    WeeklyMaintenanceStartTime: "3:02:30"
    DeploymentType: PERSISTENT_1
    DataCompressionType: LZ4

```

Exemplo de uma configuração de cluster montando um sistema de arquivos FSx for Lustre existente

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
```

Exemplo de um cluster com a pilha de software Intel HPC Platform Specification

AWS ParallelCluster 2:

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

AWS ParallelCluster 3:

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

Observações:

- A instalação do software Intel HPC Platform Specification está sujeita aos termos e condições do [Contrato de Licença de Usuário Final da Intel](#) aplicável.

Exemplo de configurações personalizadas do IAM, incluindo: perfil da instância, função da instância, políticas adicionais para instâncias e o perfil das funções lambda associadas ao 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
```

Observações:

- Para o AWS ParallelCluster 2, as configurações do IAM são aplicadas a todas as instâncias de um cluster e `additional_iam_policies` não pode ser usado em conjunto com `ec2_iam_role`.
- Para AWS ParallelCluster 3, você pode ter configurações de IAM diferentes para nós principais e de computação e até mesmo especificar configurações de IAM diferentes para cada fila de computação.

- Para AWS ParallelCluster 3, você pode usar um perfil de instância do IAM como alternativa a um perfil do IAM. InstanceProfile, InstanceRole ou AdditionalIamPolicies não podem ser configurados juntos.

Exemplo de ações de bootstrap personalizadas

AWS ParallelCluster 2:

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

AWS ParallelCluster 3:

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

```

    Args: ['R', 'curl', 'wget']
  OnNodeConfigured:
    Script: s3://bucket_name/scripts/post_install.sh
    Args: ['R', 'curl', 'wget']
  Iam:
    S3Access:
      - BucketName: bucket_name

```

Exemplo de um cluster com acesso de leitura e gravação aos recursos do bucket do S3

AWS ParallelCluster 2:

```

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

```

AWS ParallelCluster 3:

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: bucket_name
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: bucket_name
        KeyName: read_and_write/
        EnableWriteAccess: True
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue1
      ...
    Iam:
      S3Access:
        - BucketName: bucket_name
          KeyName: read_only/
          EnableWriteAccess: False
        - BucketName: bucket_name
          KeyName: read_and_write/
          EnableWriteAccess: True

```

Linguagem inclusiva

AWS ParallelCluster 3 usa as palavras "nó principal" em lugares onde "mestre" foi usado no AWS ParallelCluster 2. Essa transmissão inclui o seguinte:

- A variável exportada no ambiente de trabalho AWS Batch foi alterada: de `MASTER_IP` para `PCLUSTER_HEAD_NODE_IP`.
- Todas as saídas AWS CloudFormation foram alteradas de `Master*` para `HeadNode*`.
- Todos os `NodeType` e tags foram alterados de `Master` para `HeadNode`.

Compatibilidade com programadores

O AWS ParallelCluster 3.x não suporta programadores Son of Grid Engine (SGE) e Torque.

Os comandos AWS Batch `awsbhosts`, `awsbkill`, `awsbout`, `awsbqueues`, `awsbstat` e `awsbsub` são distribuídos como um pacote PyPI `aws-parallelcluster-awsbatch-cli` separado. Este pacote é instalado pelo AWS ParallelCluster no nó principal. Você ainda pode usar esses comandos AWS Batch a partir do nó principal do cluster. No entanto, se você deseja usar comandos AWS Batch de um local diferente do nó principal, você deve primeiro instalar o pacote PyPI `aws-parallelcluster-awsbatch-cli`.

CLI do AWS ParallelCluster

A interface de linha de comandos (CLI) do AWS ParallelCluster foi alterada. A nova sintaxe é descrita em [Comandos da CLI do AWS ParallelCluster](#). O formato de saída para a CLI é uma string [JSON](#).

Como configurar um novo cluster

O comando `pcluster configure` inclui parâmetros diferentes no AWS ParallelCluster 3 em comparação com o AWS ParallelCluster 2. Para obter mais informações, consulte [pcluster configure](#).

Observe também que a sintaxe do arquivo de configuração foi alterada em relação ao AWS ParallelCluster 2. Para obter uma referência completa das configurações do cluster, consulte [Arquivo de configuração do cluster](#).

Como criar um novo cluster

O comando `pcluster create` do AWS ParallelCluster 2 foi substituído pelo comando [pcluster create-cluster](#).

Observe que o comportamento padrão no AWS ParallelCluster 2.x, sem a opção `-nw`, é aguardar os eventos de criação do cluster, enquanto o comando do AWS ParallelCluster 3.x retorna imediatamente. O progresso da criação do cluster pode ser monitorado usando [`pcluster describe-cluster`](#).

Um arquivo de configuração AWS ParallelCluster 3 contém uma única definição de cluster, portanto, o parâmetro `-t` não é mais necessário.

Veja a seguir um exemplo do arquivo de configuração.

```
# 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
```

Como listar clusters

O comando `pcluster list` do AWS ParallelCluster 2.x deve ser substituído pelo comando [`pcluster list-clusters`](#).

Observação: você precisa da CLI do AWS ParallelCluster v2 para listar clusters criados com versões 2.x do AWS ParallelCluster. Consulte [Instalar AWS ParallelCluster em um ambiente virtual \(recomendado\)](#) para ver como instalar várias versões do AWS ParallelCluster usando ambientes virtuais.

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

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

Como iniciar e interromper um cluster

Os comandos `pcluster start` e `pcluster stop` do AWS ParallelCluster 2.x devem ser substituídos por comandos [pcluster update-compute-fleet](#).

Iniciando uma frota de computação:

```
# 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
```

Interrompendo uma frota de computação:

```
# 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
```

Conexão com um cluster

O comando `pcluster ssh` do AWS ParallelCluster 2.x tem nomes de parâmetros diferentes no AWS ParallelCluster 3.x. Consulte [pcluster ssh](#).

Conexão com um 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
```

Atualização da configuração do IMDS

A partir da versão 3.0.0, o AWS ParallelCluster introduziu compatibilidade para restringir o acesso ao IMDS do nó principal (e às credenciais do perfil de instância) a um subconjunto de superusuários, por padrão. Para obter mais informações, consulte [Propriedades do Imds](#).

Regiões compatíveis do AWS ParallelCluster

AWS ParallelCluster versão 3 está disponível nas seguintes Regiões da AWS:

Nome da Região	Região
Leste dos EUA (Ohio)	us-east-2
Leste dos EUA (N. da Virgínia)	us-east-1
Oeste dos EUA (N. da Califórnia)	us-west-1
Oeste dos EUA (Oregon)	us-west-2
África (Cidade do Cabo)	af-south-1
Ásia-Pacífico (Hong Kong)	ap-east-1

Nome da Região	Região
Ásia-Pacífico (Mumbai)	ap-south-1
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Singapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Ásia-Pacífico (Tóquio)	ap-northeast-1
Canadá (Central)	ca-central-1
China (Pequim)	cn-north-1
China (Ningxia)	cn-northwest-1
Europa (Frankfurt)	eu-central-1
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (Milão)	eu-south-1
Europa (Paris)	eu-west-3
Europa (Estocolmo)	eu-north-1
Oriente Médio (Barém)	me-south-1
América do Sul (São Paulo)	sa-east-1
AWS GovCloud (Leste dos EUA)	us-gov-east-1
AWS GovCloud (Oeste dos EUA)	us-gov-west-1
Israel (Tel Aviv)	il-central-1

Usando AWS ParallelCluster

Tópicos

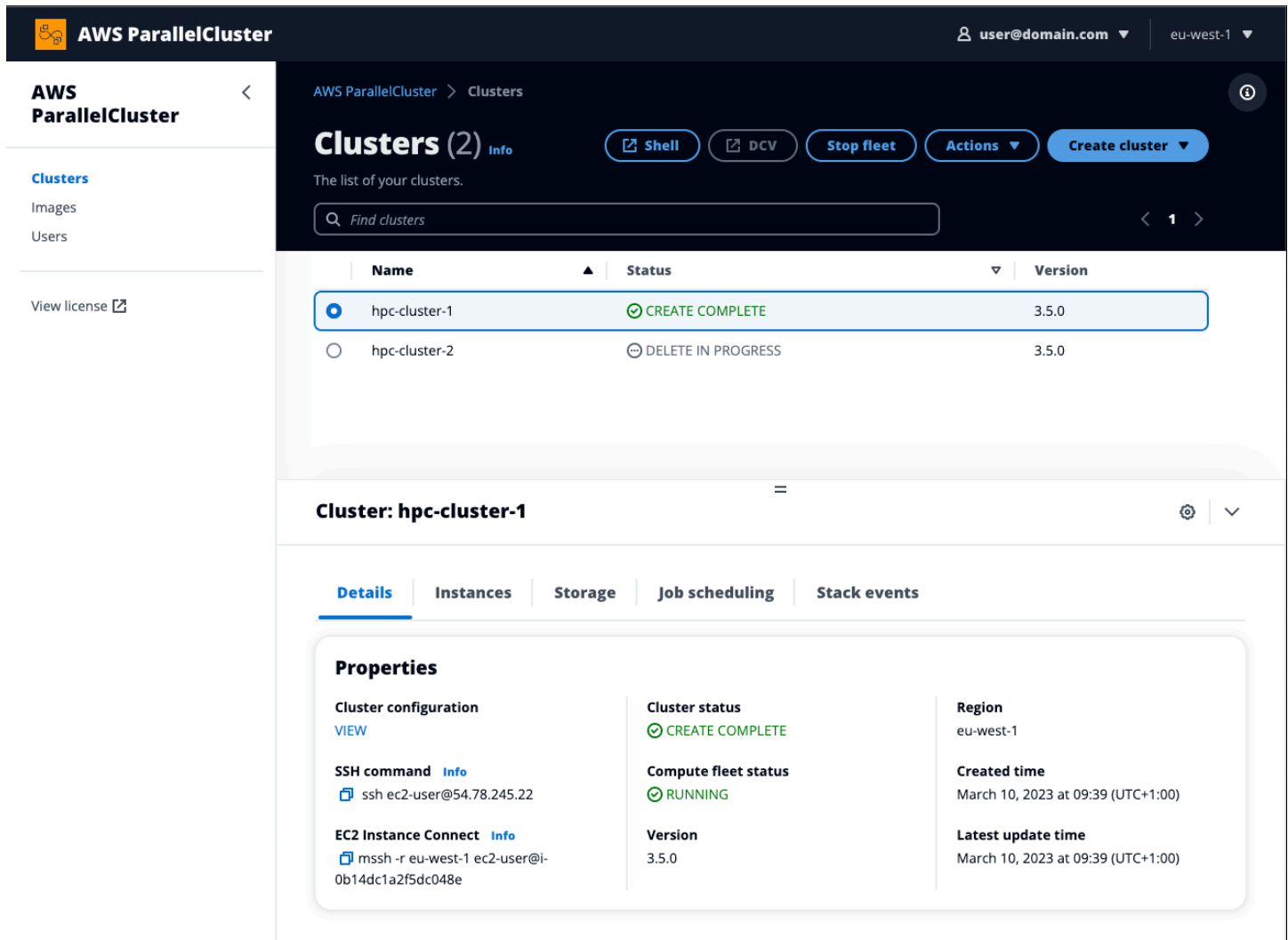
- [Interface do usuário AWS ParallelCluster](#)
- [Configuração de VPC AWS Lambda no AWS ParallelCluster](#)
- [AWS Identity and Access Management permissões em AWS ParallelCluster](#)
- [Configurações de rede](#)
- [Nós de login](#)
- [Ações de bootstrap personalizadas](#)
- [Como trabalhar com o Amazon S3](#)
- [Trabalho com Instâncias spot](#)
- [Agendadores suportados por AWS ParallelCluster](#)
- [Armazenamento compartilhado](#)
- [Recursos do AWS ParallelCluster e marcação](#)
- [Monitoramento do AWS ParallelCluster e de logs](#)
- [AWS CloudFormation recurso personalizado](#)
- [Elastic Fabric Adapter](#)
- [Habilitar o Intel MPI](#)
- [API AWS ParallelCluster](#)
- [Conecte-se ao nó principal por meio do NICE DCV](#)
- [Usar o pcluster update-cluster](#)
- [AWS ParallelCluster Personalização da AMI](#)
- [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#)
- [Aplicação de patches de AMI e substituição de instâncias do EC2](#)
- [Sistema operacional](#)

Interface do usuário AWS ParallelCluster

A IU AWS ParallelCluster é uma interface de usuário baseada na Web que serve como um painel para criar, monitorar e gerenciar clusters. Você instala e acessa a interface do usuário AWS

ParallelCluster no seu Conta da AWS. A interface do usuário AWS ParallelCluster foi adicionada com a AWS ParallelCluster versão 3.5.0.

Para instalar a interface do usuário AWS ParallelCluster e começar, consulte [Instalando a interface do usuário do AWS ParallelCluster](#) e [Configurar e criar um cluster com a interface do usuário do AWS ParallelCluster](#).



The screenshot displays the AWS ParallelCluster user interface. At the top, the header shows the AWS ParallelCluster logo, the user's email (user@domain.com), and the region (eu-west-1). The main navigation menu on the left includes 'Clusters', 'Images', and 'Users'. The central area is titled 'Clusters (2)' and contains a search bar and a table of clusters. The table has columns for Name, Status, and Version. Below the table, the details for '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, Compute fleet status, Version, Region, Created time, and Latest update time.

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

Cluster: hpc-cluster-1

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)


A UI AWS ParallelCluster oferece suporte aos seguintes recursos:

- Exibe o seguinte:
 - A lista de clusters que você criou no seu Conta da AWS com AWS ParallelCluster.
 - O status e os detalhes disponíveis para seus clusters listados.
 - O evento de pilha do CloudFormation e os logs do AWS ParallelCluster que você pode usar para monitorar.

- O status dos trabalhos que estão sendo executados em seus clusters.
- A lista de imagens personalizadas que você pode usar para criar clusters.
- A lista de imagens oficiais que a interface usa para criar clusters.
- A lista de usuários que têm acesso à interface de usuário AWS ParallelCluster. Você pode adicionar e remover usuários.
- Fornece orientação passo a passo para criar e editar (atualizar) um cluster e selecionar recursos de cluster compatíveis para adicionar, editar ou remover. Os campos de entrada inacessíveis não podem ser alterados na configuração do cluster que está sendo editada. Você tem a opção de realizar uma validação de simulação da configuração do cluster antes da implantação do cluster.
- Possui links diretos de shell para acessar o nó de cabeçalho na visualização Clusters. Escolha Add SSM session (Adicionar sessão SSM) durante a orientação passo a passo para adicionar o acesso direto ao shell e a política SSM Managed Instance Core no nó de cabeçalho.

Considere o seguinte ao utilizar a UI AWS ParallelCluster para criar e gerenciar os clusters:

- Você só pode criar e editar clusters ou criar imagens com a mesma versão AWS ParallelCluster usada para criar a interface do usuário AWS ParallelCluster. Clusters ou imagens de versões anteriores só podem ser visualizados. Se você gerencia várias versões de clusters e imagens, recomendamos criar uma instância de interface de usuário AWS ParallelCluster para oferecer suporte a cada versão.
- A interface do usuário AWS ParallelCluster foi projetada para espelhar a funcionalidade da `pccluster` CLI. Há algumas diferenças. Se você estiver de acordo com a orientação passo a passo, estará usando todos os recursos suportados. Antes da implantação, você tem a opção de editar a configuração do cluster ou da imagem manualmente. Se você fizer isso, recomendamos que valide a configuração escolhendo Simulação para verificar se suas edições são totalmente suportadas.

 Note

A UI AWS ParallelCluster não oferece suporte ao AWS Batch.

Configuração de VPC AWS Lambda no AWS ParallelCluster

O AWS ParallelCluster usa AWS Lambda para realizar operações durante o ciclo de vida do cluster. Uma [função do AWS Lambda sempre é executada dentro de uma VPC](#) de propriedade do serviço Lambda. Essa função do Lambda também pode ser conectada a sub-redes privadas em uma nuvem privada virtual (VPC) para acessar recursos privados.

Note

As funções do Lambda não podem se conectar diretamente a uma VPC com a locação de instâncias dedicadas. Para se conectar a recursos em uma VPC dedicada, emparelhe-a com uma segunda VPC com uma locação padrão que possa se conectar a uma VPC dedicada. Para obter mais informações, consulte [Instâncias dedicadas](#) no Guia do usuário do EC2 para instâncias Linux e [Como faço para conectar uma função Lambda a uma VPC dedicada?](#) da Central de Conhecimento da AWS.

As funções do Lambda criadas por AWS ParallelCluster podem ser conectadas a uma VPC privada. Essas funções do Lambda precisam ter acesso aos Serviços da AWS. Você pode fornecer acesso pela Internet ou por endpoints da VPC usando os métodos a seguir.

- Acesso à Internet

Para acessar a Internet e os Serviços da AWS, uma função do Lambda requer a conversão de endereços de rede (NAT). Defina uma rota para tráfego de saída da sub-rede privada para o [gateway NAT](#) em uma sub-rede pública.

- VPC endpoints

Vários serviços da AWS oferecem [VPC endpoints](#). É possível usar endpoints da VPC para conectar-se ao Serviços da AWS de uma VPC sem acesso à Internet. Para ver a lista de endpoints da VPC do AWS ParallelCluster, consulte [Rede](#).

Note

Cada combinação de sub-redes e grupos de segurança deve fornecer acesso ao Serviços da AWS usando um desses métodos. Os grupos de segurança e sub-redes devem estar na mesma VPC.

Para obter mais informações, consulte [endpoint da VPC](#) no Guia do usuário da Amazon Virtual Private Cloud e [Acesso à Internet e serviços para funções conectadas à VPC](#) no Guia do desenvolvedor AWS Lambda.

Para configurar o uso de VPCs e funções do Lambda, consulte [DeploymentSettings / LambdaFunctionsVpcConfig](#) para clusters ou [DeploymentSettings / LambdaFunctionsVpcConfig](#) para imagens.

AWS Identity and Access Management permissões em AWS ParallelCluster

AWS ParallelCluster usa permissões do IAM para controlar o acesso aos recursos ao criar e gerenciar clusters.

Para criar e gerenciar clusters em uma AWS conta, são AWS ParallelCluster necessárias permissões em dois níveis:

- Permissões que o usuário `pcluster` precisa para invocar os comandos `pcluster` da CLI para criar e gerenciar clusters.
- Permissões que os recursos do cluster exigem para realizar ações do cluster.

AWS ParallelCluster usa um [perfil e uma função de instância do EC2](#) para fornecer permissões de recursos de cluster. Para gerenciar as permissões dos recursos do cluster, AWS ParallelCluster também são necessárias permissões para os recursos do IAM. Para ter mais informações, consulte [AWS ParallelCluster políticas de exemplo de usuário para gerenciar recursos do IAM](#).

Os usuários do `pcluster` exigem permissões do IAM para usar a CLI `pcluster` para criar e gerenciar um cluster e seus recursos. Essas permissões estão incluídas nas políticas do IAM que podem ser adicionadas a um usuário ou perfil. Para obter mais informações sobre perfis do IAM, consulte [Criar um perfil de usuário](#) no AWS Identity and Access Management Guia do usuário do IAM.

Você também pode usar [AWS ParallelCluster parâmetros de configuração para gerenciar permissões do IAM](#).

As seções a seguir contêm as permissões necessárias com exemplos.

Para usar as políticas de exemplo, substitua `<REGION>`, `<AWS ACCOUNT ID>` e cadeias de caracteres semelhantes pelos valores apropriados.

Os exemplos de políticas a seguir incluem nomes de recursos da Amazon (ARNs) para os recursos. Se você estiver trabalhando nas partições AWS GovCloud (US) ou na AWS China, os ARNs devem ser alterados. Especificamente, eles devem ser alterados de “arn:aws” para “arn:aws-us-gov” para a AWS GovCloud (US) partição ou “arn:aws-cn” para a partição da China. AWS Para obter mais informações, consulte [Amazon Resource Names \(ARNs\) nas AWS GovCloud \(US\) regiões](#) no Guia AWS GovCloud (US) do usuário e [ARNs para AWS serviços na China](#) em Introdução aos AWS serviços na China.

Você pode acompanhar as alterações nas políticas de exemplo na [AWS ParallelCluster documentação em GitHub](#).

Tópicos

- [AWS ParallelCluster Funções de instância do EC2](#)
- [AWS ParallelCluster exemplo de políticas pcluster de usuário](#)
- [AWS ParallelCluster políticas de exemplo de usuário para gerenciar recursos do IAM](#)
- [AWS ParallelCluster parâmetros de configuração para gerenciar permissões do IAM](#)

AWS ParallelCluster Funções de instância do EC2

Quando você cria um cluster com as configurações padrão, AWS ParallelCluster usa [perfis de instância](#) EC2 para criar automaticamente uma [função de instância](#) EC2 de cluster padrão que fornece as permissões necessárias para criar e gerenciar o cluster e seus recursos.

Alternativas ao uso da função de AWS ParallelCluster instância padrão

No lugar da função de AWS ParallelCluster instância padrão, você pode usar a definição de configuração do `InstanceRole` cluster para especificar sua própria função do IAM existente para o EC2. Para ter mais informações, consulte [AWS ParallelCluster parâmetros de configuração para gerenciar permissões do IAM](#). Normalmente, você especifica os perfis do IAM existentes para controlar totalmente as permissões concedidas ao EC2.

Se sua intenção é adicionar políticas extras à função de instância padrão, recomendamos que você transmita as políticas adicionais do IAM com a configuração `AdditionalIamPolicies` em vez de usar as configurações `InstanceProfile` ou `InstanceRole`. Você pode atualizar `AdditionalIamPolicies` quando atualizar seu cluster, mas não pode atualizar `InstanceRole` quando atualizar seu cluster.

AWS ParallelCluster exemplo de políticas **pcluster** de usuário

Os exemplos a seguir mostram as políticas de usuário necessárias para criar AWS ParallelCluster e gerenciar seus recursos usando a `pcluster` CLI. Agora você pode anexar a política a um usuário ou um perfil.

Tópicos

- [Política básica de usuário `pcluster` do AWS ParallelCluster](#)
- [Política de usuário adicional `pcluster` do AWS ParallelCluster ao usar o programador AWS Batch](#)
- [Política de usuário adicional `pcluster` do AWS ParallelCluster ao usar o Amazon FSx para Lustre](#)
- [AWS ParallelCluster política de `pcluster` usuário de criação de imagens](#)

Política básica de usuário **pcluster** do AWS ParallelCluster

A política a seguir mostra as permissões necessárias para executar AWS ParallelCluster `pcluster` comandos.

A última ação listada na política está incluída para fornecer validação de todos os segredos especificados na configuração do cluster. Por exemplo, um AWS Secrets Manager segredo é usado para configurar a [DirectoryService](#) integração. Nesse caso, um cluster é criado somente se existir um segredo válido em [PasswordSecretArn](#). Se essa ação for omitida, a validação secreta será ignorada. Para melhorar sua postura de segurança, recomendamos que você defina o escopo dessa declaração de política adicionando somente os segredos especificados na configuração do cluster.

Note

Se os sistemas de arquivos Amazon EFS existentes forem os únicos sistemas de arquivos usados em seu cluster, você poderá definir o escopo dos exemplos de declarações de política do Amazon EFS para os sistemas de arquivos específicos referenciados no arquivo [Seção SharedStorage](#) de configuração do cluster.

```
{
  "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"
  ],
  "Resource": "arn:aws:dynamodb:*:<AWS ACCOUNT ID>: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"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatch"
},
{
    "Action": [
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetPolicy",
        "iam:SimulatePrincipalPolicy",
        "iam:GetInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/*",
        "arn:aws:iam::aws:policy/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRead"
},
{
    "Action": [
        "iam:CreateInstanceProfile",
        "iam>DeleteInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>: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::<AWS ACCOUNT ID>: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:*:<AWS ACCOUNT ID>:function:parallelcluster-*",
        "arn:aws:lambda:*:<AWS ACCOUNT ID>: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:*:<AWS ACCOUNT ID>:*"
    ],
    "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"
    ],
    "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:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET
NAME>",
    "Effect": "Allow"
  }
]
}

```

Política de usuário adicional **pcluster** do AWS ParallelCluster ao usar o programador AWS Batch

Caso você precise criar e gerenciar um cluster com AWS Batch agendador, a seguinte política adicional é necessária.

```

{
  "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::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": [
          "batch.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:CreateServiceLinkedRole",
      "iam>DeleteServiceLinkedRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/
batch.amazonaws.com/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "codebuild:*"
    ],
    "Resource": "arn:aws:codebuild:*:<AWS ACCOUNT ID>: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"
}
]
}

```

Política de usuário adicional **pcluster** do AWS ParallelCluster ao usar o Amazon FSx para Lustre

Caso você precise criar e gerenciar um cluster com o Amazon FSx para Lustre, a seguinte política adicional é necessária.

Note

Se os sistemas de arquivos Amazon FSx existentes forem os únicos sistemas de arquivos usados em seu cluster, você poderá definir o escopo dos exemplos de declarações de política do Amazon FSx para os sistemas de arquivos específicos referenciados no arquivo [Seção SharedStorage](#) de configuração do cluster.

```
{
```

```

"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:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "FSx"
  },
  {
    "Action": [
      "iam:CreateServiceLinkedRole",
      "iam:AttachRolePolicy",
      "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
    ],
  },
]

```

```

    "Resource": "arn:aws:s3:::<S3 NAME>",
    "Effect": "Allow"
  }
]
}

```

AWS ParallelCluster política de **pcluster** usuário de criação de imagens

Os usuários que pretendem criar imagens EC2 personalizadas com AWS ParallelCluster devem ter o seguinte conjunto de permissões.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeImages",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeInstanceTypes",
        "ec2:DeregisterImage",
        "ec2:DeleteSnapshot"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2"
    },
    {
      "Action": [
        "iam:CreateInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/ParallelClusterImage*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IAM"
    }
  ],
}

```



```

    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "lambda.amazonaws.com",
            "ec2.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IAMPassRole"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs>DeleteLogGroup"
      ],
      "Resource": [
        "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/imagebuilder/
ParallelClusterImage-*",
        "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/lambda/
ParallelClusterImage-*"
      ],
      "Effect": "Allow",
      "Sid": "CloudWatch"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation>CreateStack",
        "cloudformation>DeleteStack"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:<AWS ACCOUNT ID>: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:*:<AWS ACCOUNT ID>: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:*:<AWS ACCOUNT ID>:image/parallelclusterimage-*",

```

```

        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:component/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:distribution-configuration/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>: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:*:<AWS ACCOUNT ID>: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 políticas de exemplo de usuário para gerenciar recursos do IAM

Ao usar AWS ParallelCluster para criar clusters ou AMIs personalizadas, é necessário fornecer políticas do IAM que contenham permissões para conceder o conjunto necessário de permissões aos AWS ParallelCluster componentes. Esses recursos do IAM podem ser criados automaticamente AWS ParallelCluster ou fornecidos como entrada ao criar um cluster ou uma imagem personalizada.

Você pode usar os modos a seguir para fornecer ao AWS ParallelCluster usuário as permissões necessárias para acessar os recursos do IAM usando políticas adicionais do IAM na configuração.

Tópicos

- [Modo de acesso privilegiado do IAM](#)
- [Modo de acesso restrito do IAM](#)
- [Modo PermissionsBoundary](#)

Modo de acesso privilegiado do IAM

Com esse modo, cria AWS ParallelCluster automaticamente todos os recursos do IAM necessários. Essas políticas do IAM têm um escopo reduzido para permitir o acesso somente aos recursos do cluster.

Para ativar o modo de acesso privilegiado do IAM, adicione a política a seguir à função do usuário.

Note

Se você configurar [AdditionalPolicies](#) parâmetros [HeadNodeIam/Scheduling/AdditionalPolicies](#) ou [SlurmQueues/Iam//](#), deverá fornecer ao AWS ParallelCluster usuário permissão para anexar e desanexar políticas de função para cada política adicional, conforme mostrado na política a seguir. Adicione os ARNs de política adicionais à condição para anexar e desanexar políticas de função.

Warning

Esse modo permite que o usuário tenha privilégios de administrador do IAM no Conta da AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:TagRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRole"
    },
    {
      "Action": [
```

```

        "iam:CreateRole"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamCreateRole"
},
{
    "Action": [
        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
},
{
    "Condition": {
        "ArnLike": {
            "iam:PolicyARN": [
                "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
                "arn:aws:iam::<AWS ACCOUNT ID>: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::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamPolicy"
  }
]
}

```

Modo de acesso restrito do IAM

Quando nenhuma política adicional do IAM é concedida ao usuário, os perfis do IAM exigidos pelos clusters ou pela criação de imagens personalizadas precisam ser criados manualmente por um administrador e passados como parte da configuração do cluster.

Ao criar um cluster, os seguintes parâmetros são obrigatórios:

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

Ao criar uma imagem personalizada, os seguintes parâmetros são obrigatórios:

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

Os perfis do IAM passados como parte dos parâmetros listados acima devem ser criados no prefixo do caminho `/parallelcluster/`. Se isso não for possível, a política do usuário precisará ser atualizada para conceder permissão `iam:PassRole` sobre as funções personalizadas específicas, como no exemplo a seguir.

```

{
  "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

Atualmente, esse modo não permite o gerenciamento de AWS Batch clusters porque nem todas as funções do IAM podem ser passadas na configuração do cluster.

Modo **PermissionsBoundary**

Esse modo delega à AWS ParallelCluster criação de funções do IAM que estão vinculadas ao limite de permissões configurado do IAM. Para obter mais informações sobre limites de permissões do IAM, consulte [Limites de permissões para entidades do IAM](#) no Guia do usuário do IAM.

A política a seguir precisa ser adicionada à função do usuário.

Na política, substitua < *permissions-boundary-arn* > pelo ARN da política do IAM a ser aplicado como limite de permissões.

Warning

Se você configurar os parâmetros [HeadNode](#) / [Iam](#) / [AdditionalPolicies](#) ou [Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalPolicies](#), deverá conceder ao usuário permissão para anexar e desanexar políticas de função para cada política adicional, conforme mostrado na política a seguir. Adicione os ARNs de política adicionais à condição para anexar e desanexar políticas de função.

```

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

```



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

```

    "Sid": "IamInlinePolicy"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          <permissions-boundary-arn>
        ]
      },
      "ArnLike": {
        "iam:PolicyARN": [
          "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
          "arn:aws:iam::<AWS ACCOUNT ID>: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::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamPolicy"
  }
]
}

```

Quando esse modo está ativado, você deve especificar o limite de permissões ARN no parâmetro de configuração [Iam / PermissionsBoundary](#) ao criar ou atualizar um cluster e no parâmetro [Build / Iam / PermissionBoundary](#) ao criar uma imagem personalizada.

AWS ParallelCluster parâmetros de configuração para gerenciar permissões do IAM

AWS ParallelCluster expõe uma série de opções de configuração para personalizar e gerenciar as permissões e funções do IAM que são usadas em um cluster ou durante o processo de criação da AMI personalizada.

Tópicos

- [Configuração do cluster](#)
- [Configuração de imagem personalizada](#)

Configuração do cluster

Tópicos

- [Perfil do IAM para o nó principal](#)
- [Acesso do Amazon S3](#)
- [Política do IAM adicionais](#)
- [AWS Lambda função de funções](#)
- [Perfil do IAM para nós de computação](#)
- [Limite de permissões](#)

Perfil do IAM para o nó principal

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

Com essa opção, você substitui o perfil padrão do IAM atribuído ao nó principal do cluster. Para obter detalhes adicionais, consulte a referência [InstanceProfile](#).

Aqui está o conjunto mínimo de políticas a serem usadas como parte dessa função quando o programador é o Slurm:

- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`. Para obter mais informações, consulte [Criar funções e usuários do IAM para uso com o CloudWatch agente](#) no Guia CloudWatch do usuário da Amazon.
- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`. Para obter mais informações, consulte [Políticas gerenciadas pela AWS para AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager .
- Política do IAM adicional:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.<REGION>/*",
        "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:<REGION>:<AWS ACCOUNT ID>: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::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "arn:aws:iam::<AWS ACCOUNT ID>: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:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>: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:<REGION>:<AWS ACCOUNT
ID>:secret:<SECRET_ID>",
    "Effect": "Allow"
}
]
}

```

Observe que, caso [Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) seja usado para substituir o perfil de computação do IAM, a política do nó principal relatada acima precisa incluir esse perfil na seção Resource da permissão iam:PassRole.

Aqui está o conjunto mínimo de políticas a serem usadas como parte dessa função quando o programador é o AWS Batch:

- Política do IAM gerenciada pela arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy. Para obter mais informações, consulte [Criar funções e usuários do IAM para uso com o CloudWatch agente](#) no Guia CloudWatch do usuário da Amazon.

- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`. Para obter mais informações, consulte [Políticas gerenciadas pela AWS para AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager .
- Política do IAM adicional:

```
{
  "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.<REGION>/*",
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>: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:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-stream:PclusterJobDefinition*",
      "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/AWSBatch-PclusterComputeEnviron*",
      "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/AWSBatch-Pcluster*",
      "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/PclusterJobQueue*",
      "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-definition/PclusterJobDefinition*:*",
      "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeVolumes",
      "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
},

```



```

    {
      "Action": [
        "ec2:CreateTags",
        "ec2:AttachVolume"
      ],
      "Resource": [
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStacks",
        "cloudformation:SignalResource"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET_ID>",
      "Effect": "Allow"
    }
  ]
}

```

Acesso do Amazon S3

[HeadNode](#) / [Iam](#) / [S3Access](#) ou [Scheduling](#) / [SlurmQueues](#) / [S3Access](#)

Nessas seções de configuração, você pode personalizar o acesso ao Amazon S3 concedendo políticas adicionais do Amazon S3 aos perfis do IAM associados ao nó principal ou aos nós de computação do cluster quando esses perfis são criados pelo AWS ParallelCluster. Para obter mais informações, consulte a documentação de referência para cada parâmetro de configuração.

Esse parâmetro só pode ser usado quando o usuário está configurado com [Modo de acesso privilegiado do IAM](#) ou [Modo PermissionsBoundary](#).

Política do IAM adicionais

[HeadNode / Iam / AdditionalIamPolicies](#) ou [SlurmQueues / Iam / AdditionalIamPolicies](#)

Use essa opção para anexar políticas gerenciadas adicionais do IAM às funções do IAM associadas ao nó principal ou aos nós de computação do cluster quando essas funções são criadas por AWS ParallelCluster.

Warning

Para usar essa opção, certifique-se de que o [usuário do AWS ParallelCluster](#) tenha as permissões `iam:AttachRolePolicy` e `iam:DetachRolePolicy` para as políticas do IAM que precisam ser anexadas.

AWS Lambda função de funções

[Iam / Roles / LambdaFunctionsRole](#)

Essa opção substitui a função associada a todas as AWS Lambda funções usadas durante o processo de criação do cluster. AWS Lambda precisa ser configurado como o principal autorizado a assumir a função.

Note

Se [DeploymentSettings / LambdaFunctionsVpcConfig](#) estiver definido, `LambdaFunctionsRole` deverá incluir a [permissão de perfil do AWS Lambda](#) para definir a configuração da VPC.

Aqui está o conjunto mínimo de políticas a serem usadas como parte desse perfil:

```
{
  "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:<REGION>:<AWS ACCOUNT ID>: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/*"
    ]
  }
]
}

```

Perfil do IAM para nós de computação

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

Essa opção permite substituir o perfil do IAM que é atribuído aos nós de computação do cluster. Para ter mais informações, consulte [InstanceProfile](#).

Aqui está o conjunto mínimo de políticas a serem usadas como parte desse perfil:

- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`. Para obter mais informações, consulte [Criar funções e usuários do IAM para uso com o CloudWatch agente](#) no Guia CloudWatch do usuário da Amazon.
- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`. Para obter mais informações, consulte [Políticas gerenciadas pela AWS para AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager .
- Política do IAM adicional:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:Query",
        "dynamodb:UpdateItem",
        "dynamodb:PutItem",
        "dynamodb:GetItem"
      ],
      "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "ec2:DescribeInstanceAttribute",
      "Resource": "*",
      "Effect": "Allow"
    },
    {
```

```
    "Action": "cloudformation:DescribeStackResource",
    "Resource":
      [ "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*" ],
    "Effect" "Allow"
  }
]
```

Limite de permissões

[Iam / PermissionsBoundary](#)

Esse parâmetro força AWS ParallelCluster a anexação da política do IAM especificada como `PermissionsBoundary` a a todas as funções do IAM criadas como parte de uma implantação de cluster.

Consulte [Modo PermissionsBoundary](#) para obter a lista de políticas exigidas pelo usuário quando essa configuração é definida.

Configuração de imagem personalizada

Tópicos

- [Perfil de instância para o EC2 Image Builder](#)
- [AWS Lambda função de limpeza](#)
- [Política do IAM adicionais](#)
- [Limite de permissões](#)

Perfil de instância para o EC2 Image Builder

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

Com essa opção, você substitui o perfil do IAM atribuído à instância do EC2 iniciada pelo EC2 Image Builder para criar uma AMI personalizada.

Aqui está o conjunto mínimo de políticas a serem usadas como parte desse perfil:

- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`. Para obter mais informações, consulte [Políticas](#)

[gerenciadas pela AWS para AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager .

- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder`. Para obter mais informações, consulte [Política do EC2InstanceProfileForImageBuilder](#) no Guia do usuário do Image Builder.
- Política do IAM adicional:

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

AWS Lambda função de limpeza

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

Essa opção substitui a função associada a todas as AWS Lambda funções usadas durante o processo de criação da imagem personalizada. AWS Lambda precisa ser configurado como o principal autorizado a assumir a função.

Note

Se [DeploymentSettings](#) / [LambdaFunctionsVpcConfig](#) estiver definido, `CleanupLambdaRole` deverá incluir a [permissão de perfil do AWS Lambda](#) para definir a configuração da VPC.

Aqui está o conjunto mínimo de políticas a serem usadas como parte desse perfil:

- Política do IAM gerenciada pela `arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole`. Para obter mais informações, consulte [Políticas gerenciadas pela AWS para recursos do Lambda](#) no Guia do desenvolvedor do AWS Lambda .
- Política do IAM adicional:

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

```

```

    "Action": "imagebuilder:DeleteImageRecipe",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*/**",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteDistributionConfiguration",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:distribution-
configuration/parallelclusterimage-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder:DeleteImage",
      "imagebuilder:GetImage",
      "imagebuilder:CancelImageCreation"
    ],
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image/
parallelclusterimage-*/**",
    "Effect": "Allow"
  },
  {
    "Action": "cloudformation:DeleteStack",
    "Resource": "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/**/*",
    "Effect": "Allow"
  },
  {
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:<REGION>::image/*",
    "Effect": "Allow"
  },
  {
    "Action": "tag:TagResources",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:DeleteFunction",
      "lambda:RemovePermission"
    ],
    "Resource": "arn:aws:lambda:<REGION>:<AWS ACCOUNT
ID>:function:ParallelClusterImage-*",
    "Effect": "Allow"
  }

```



```
    },
    {
      "Action": "logs:DeleteLogGroup",
      "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/
lambda/ParallelClusterImage-*:*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS>DeleteTopic",
        "SNS:GetSubscriptionAttributes",
        "SNS:Unsubscribe"
      ],
      "Resource": "arn:aws:sns:<REGION>:<AWS ACCOUNT ID>:ParallelClusterImage-
**",
      "Effect": "Allow"
    }
  ]
}
```

Política do IAM adicionais

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

Use essa opção para anexar políticas gerenciadas do IAM adicionais ao perfil associado à instância do EC2 usada pelo EC2 Image Builder para produzir a AMI personalizada.

Warning

Para usar essa opção, certifique-se de que o [usuário do AWS ParallelCluster](#) tenha as permissões `iam:AttachRolePolicy` e `iam:DetachRolePolicy` para as políticas do IAM que precisam ser anexadas.

Limite de permissões

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

Esse parâmetro força AWS ParallelCluster a anexação da política do IAM especificada como a `PermissionsBoundary` a todas as funções do IAM criadas como parte da criação personalizada da AMI.

Consulte em [Modo `PermissionsBoundary`](#) a lista de políticas necessárias para usar essa funcionalidade.

Configurações de rede

AWS ParallelCluster usa a Amazon Virtual Private Cloud (VPC) para redes. A VPC fornece uma plataforma de rede flexível e configurável onde você pode implantar clusters.

A VPC deve ter as opções `DNS Resolution = yes`, `DNS Hostnames = yes` e DHCP com o nome de domínio correto para a Região. O conjunto de opções DHCP padrão já especifica o `AmazonProvidedDNS` necessário. Se estiver especificando mais de um servidor de nomes de domínio, consulte [Conjuntos de opções DHCP](#) no Amazon VPC User Guide.

O AWS ParallelCluster oferece suporte para as seguintes configurações de alto nível:

- Uma sub-rede para nós principais e de computação.
- Duas sub-redes, com o nó principal em uma sub-rede pública, e nós de computação em uma sub-rede privada. As sub-redes podem ser novas ou existentes.

Todas essas configurações podem operar com ou sem endereçamento IP público. O AWS ParallelCluster também pode ser implantado para usar um proxy HTTP para todas as solicitações da AWS. As combinações dessas configurações resultam em muitos cenários de implantação. Por exemplo, você pode configurar uma única sub-rede pública com todo o acesso pela Internet. Ou você pode configurar uma rede totalmente privada usando AWS Direct Connect e um proxy HTTP para todo o tráfego.

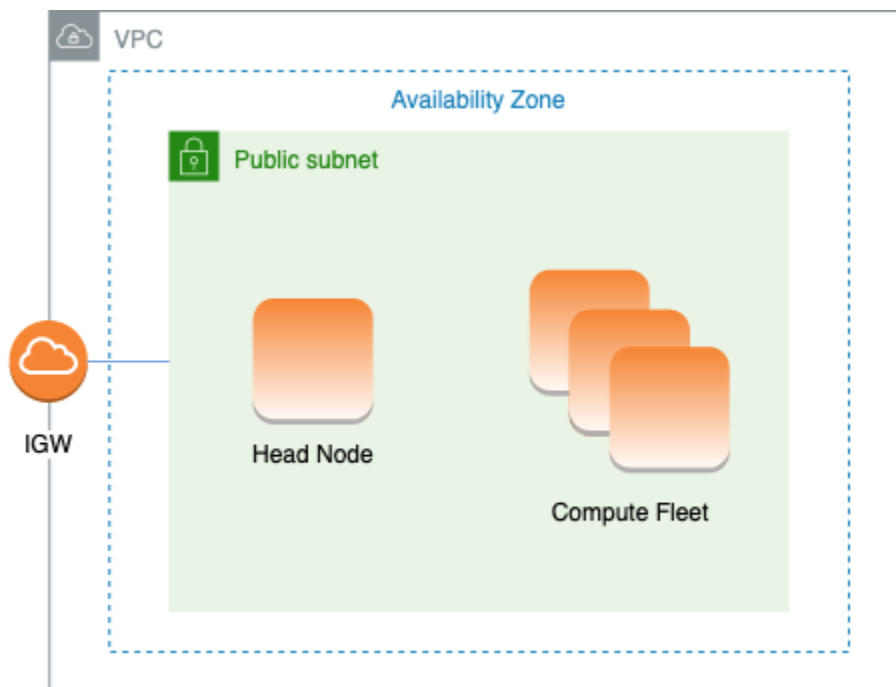
A partir do AWS ParallelCluster 3.0.0 é possível definir configurações diferentes para `SecurityGroups`, `AdditionalSecurityGroups` e `PlacementGroup` em cada fila. Para obter mais informações, consulte [HeadNode / Networking](#) e [SlurmQueues / Networking](#) e [AwsBatchQueues / Networking](#).

Consulte os seguintes diagramas de arquitetura para obter ilustrações de alguns desses cenários de rede:

Tópicos

- [O AWS ParallelCluster em uma única sub-rede pública](#)
- [O AWS ParallelCluster com duas sub-redes](#)
- [O AWS ParallelCluster em uma única sub-rede privada conectado usando o AWS Direct Connect](#)
- [AWS ParallelCluster com programador AWS Batch](#)
- [AWS ParallelCluster em uma sub-rede única sem acesso à internet](#)

O AWS ParallelCluster em uma única sub-rede pública



A configuração para essa arquitetura requer as seguintes definições:

```
# 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:
```

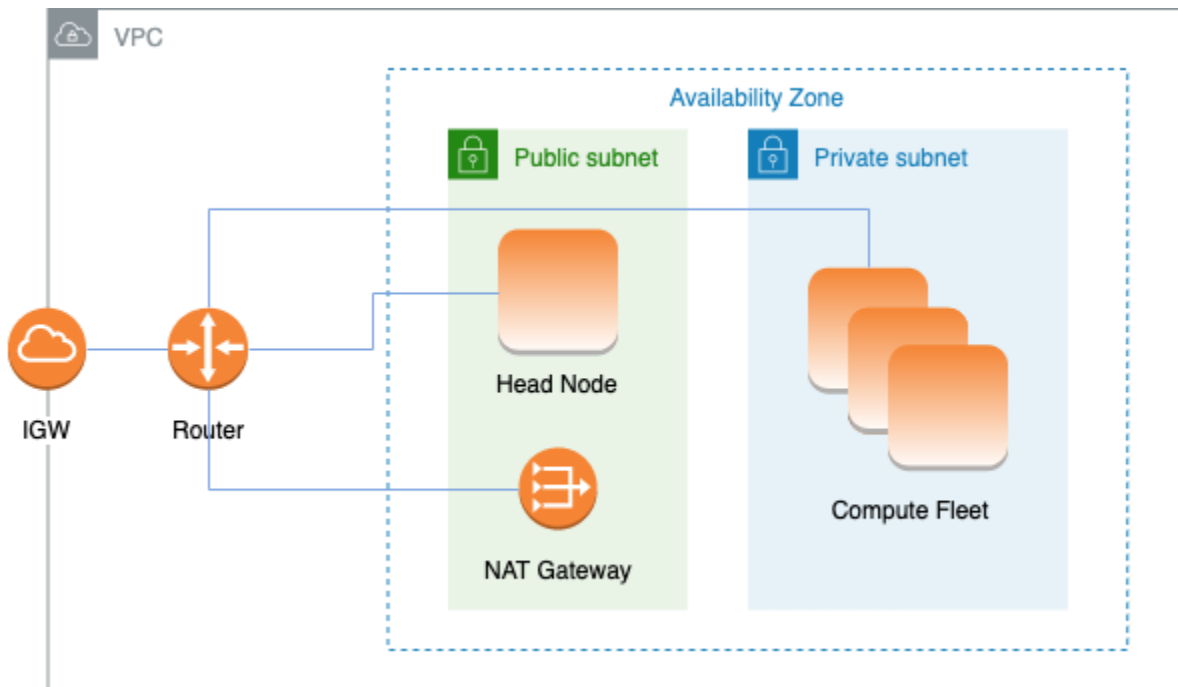
```
- subnet-12345678 # subnet with internet gateway
#AssignPublicIp: true
```

Nessa configuração, todas as instâncias do cluster devem receber um IP público para obter acesso à Internet. Para conseguir isso, faça o seguinte:

- Certifique-se de que o nó principal tenha um endereço IP público ativando a configuração "Ativar atribuição automática de endereço IPv4 público" para a sub-rede usada em [HeadNode / Networking / SubnetId](#) ou atribuindo um IP elástico em [HeadNode / Networking / ElasticIp](#).
- Certifique-se de que o nó de computação tenha um endereço IP público atribuído ativando a configuração "Ativar atribuição automática de endereço IPv4 público" para a sub-rede usada em [Scheduling / SlurmQueues / Networking / SubnetIds](#) ou atribuindo `AssignPublicIp: true` em [Scheduling / SlurmQueues / Networking](#).
- Se você definir um tipo de instância p4d ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede no nó principal, você deve definir [HeadNode / Networking / ElasticIp](#) para `true` para fornecer acesso público. IPs públicos da AWS só podem ser atribuídos a instâncias iniciadas com uma única interface de rede. Nesse caso, recomendamos que você use um [gateway NAT](#) para fornecer acesso público aos nós de computação do cluster. Para obter mais informações sobre endereços IP, consulte [Atribuir um endereço IPv4 público durante a execução de uma instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.
- Você não pode definir um tipo de instância p4d ou hp6id, ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede nos nós de computação porque os IPs públicos da AWS só podem ser atribuídos a instâncias iniciadas com uma única interface de rede. Para obter mais informações sobre endereços IP, consulte [Atribuir um endereço IPv4 público durante a execução de uma instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Para obter mais informações, consulte [Habilitar o acesso à Internet](#) no Manual do usuário da Amazon VPC.

O AWS ParallelCluster com duas sub-redes



A configuração para usar uma sub-rede privada existente para instâncias de computação requer as seguintes definições:

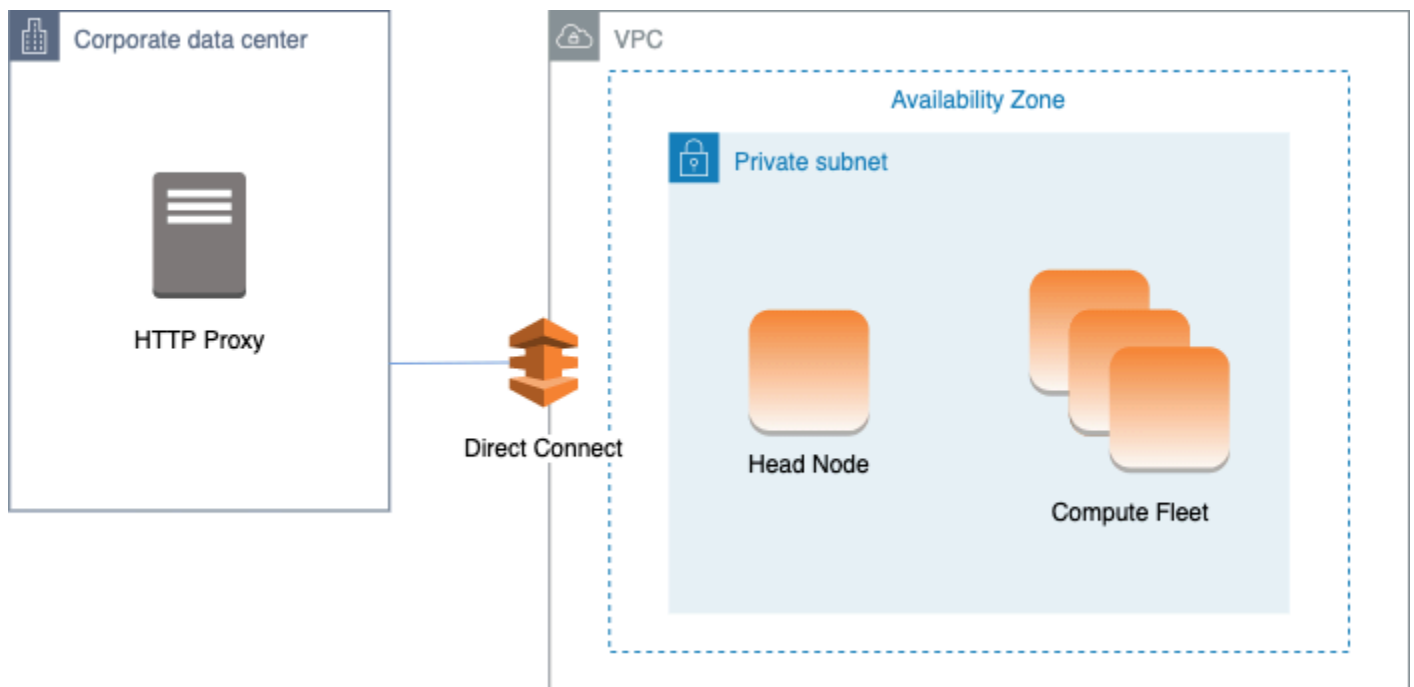
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - ...
    Networking:
      SubnetIds:
        - subnet-23456789 # subnet with NAT gateway
      #AssignPublicIp: false
```

Nessa configuração, somente o nó principal do cluster precisa ter um IP público atribuído. Você pode fazer isso ativando a configuração "Ativar atribuição automática de endereço IPv4 público" para a sub-rede usada em [HeadNode](#) / [Networking](#) / [SubnetId](#) ou atribuindo um IP elástico em [HeadNode](#) / [Networking](#) / [ElasticIp](#).

Se você definir um tipo de instância p4d ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede no nó principal, você deve definir [HeadNode / Networking / ElasticIp](#) para true para fornecer acesso público. IPs públicos da AWS só podem ser atribuídos a instâncias iniciadas com uma única interface de rede. Para obter mais informações sobre endereços IP, consulte [Atribuir um endereço IPv4 público durante a execução de uma instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Essa configuração requer um [gateway NAT](#) ou um proxy interno na sub-rede usada para as filas, para dar acesso à Internet às instâncias de computação.

O AWS ParallelCluster em uma única sub-rede privada conectado usando o AWS Direct Connect



A configuração para essa arquitetura requer as seguintes definições:

```
# 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
```

Quando [Scheduling](#) / [SlurmQueues](#) / [Networking](#) / [AssignPublicIp](#) for definido para false, como as sub-redes deverão ser configuradas corretamente para usar o Proxy para todo o tráfego. O acesso à Web é necessário tanto para os nós principais quanto para os nós de computação.

AWS ParallelCluster com programador AWS Batch

Ao usar o `awsbatch` como o tipo de programador, o AWS ParallelCluster cria um ambiente de computação gerenciado do AWS Batch. O ambiente AWS Batch gerencia as instâncias de contêiner do Amazon Elastic Container Service (Amazon ECS). Essas instâncias são executadas na sub-rede configurada no parâmetro [AwsBatchQueues](#) / [Networking](#) / [SubnetIds](#). Para que o AWS Batch funcione corretamente, as instâncias de contêiner do Amazon ECS precisam de acesso à rede externa para se comunicarem com o endpoint de serviço do Amazon ECS. Isso se converte nas seguintes situações:

- O ID de sub-rede especificado para a fila usa um [gateway NAT](#) para acessar a Internet. Recomendamos essa abordagem.
- As instâncias executadas na sub-rede da fila têm endereços IP públicos e podem acessar a Internet por meio de um gateway da Internet.

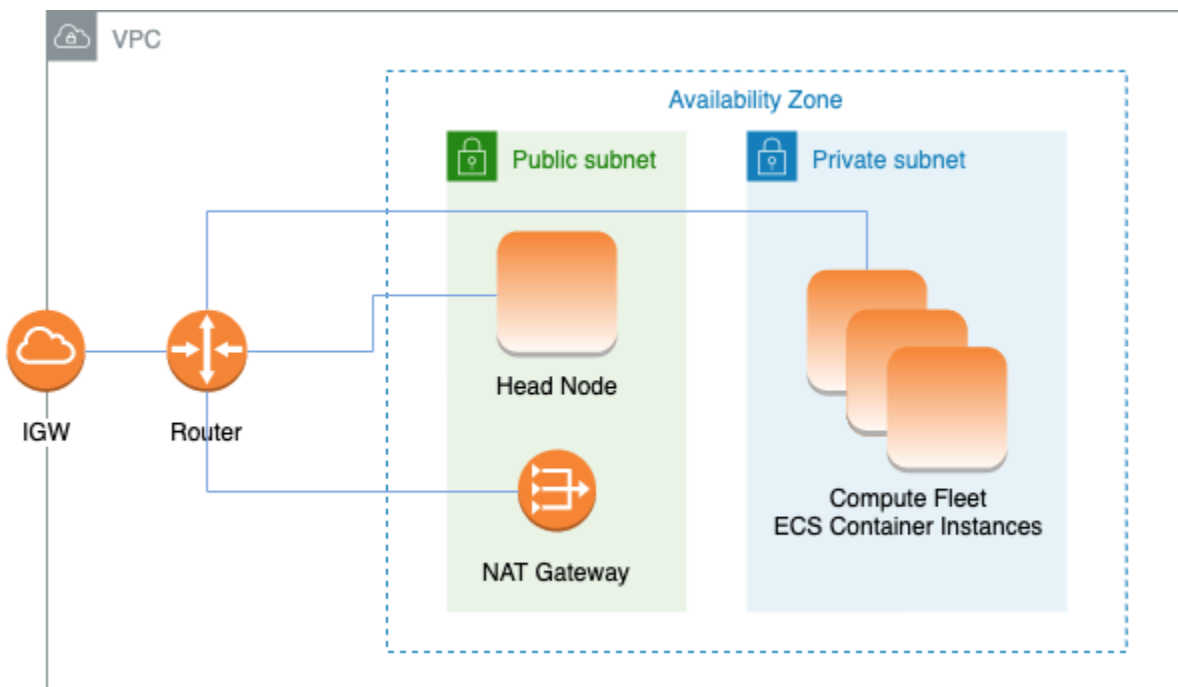
Além disso, se você estiver interessado em trabalhos em paralelo de vários nós (dos [documentos do AWS Batch](#)):

Trabalhos AWS Batch paralelos de vários nós usam o modo de rede `awsvpc` do Amazon ECS. Isso dá aos seus contêineres de trabalhos paralelos de vários nós as mesmas propriedades de redes que as instâncias do Amazon EC2. Cada contêiner de tarefa em paralelo de vários nós obtém sua própria interface de rede elástica, um endereço IP privado primário e um nome de host DNS interno. A interface de rede é criada na mesma sub-rede Amazon VPC que seu recurso de computação

do host. Todos os grupos de segurança aplicados aos seus recursos de computação também são aplicados a ele.

Ao usar redes de trabalho da Rede de tarefas do Amazon ECS, o modo de rede `aws-vpc` não fornece interfaces de rede elásticas com endereços IP públicos para as tarefas que usam o tipo de inicialização do Amazon EC2. Para acessar a Internet, as tarefas que usam o tipo de execução do Amazon EC2 devem ser executadas em uma sub-rede privada que esteja configurada para usar um gateway NAT.

Você deve configurar um [gateway NAT](#) para permitir que o cluster execute trabalhos paralelos de vários nós.



Todas as configurações e considerações anteriores também são válidas para o AWS Batch. Veja a seguir um exemplo de uma configuração personalizada do AWS Batch.

```
# 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:
```



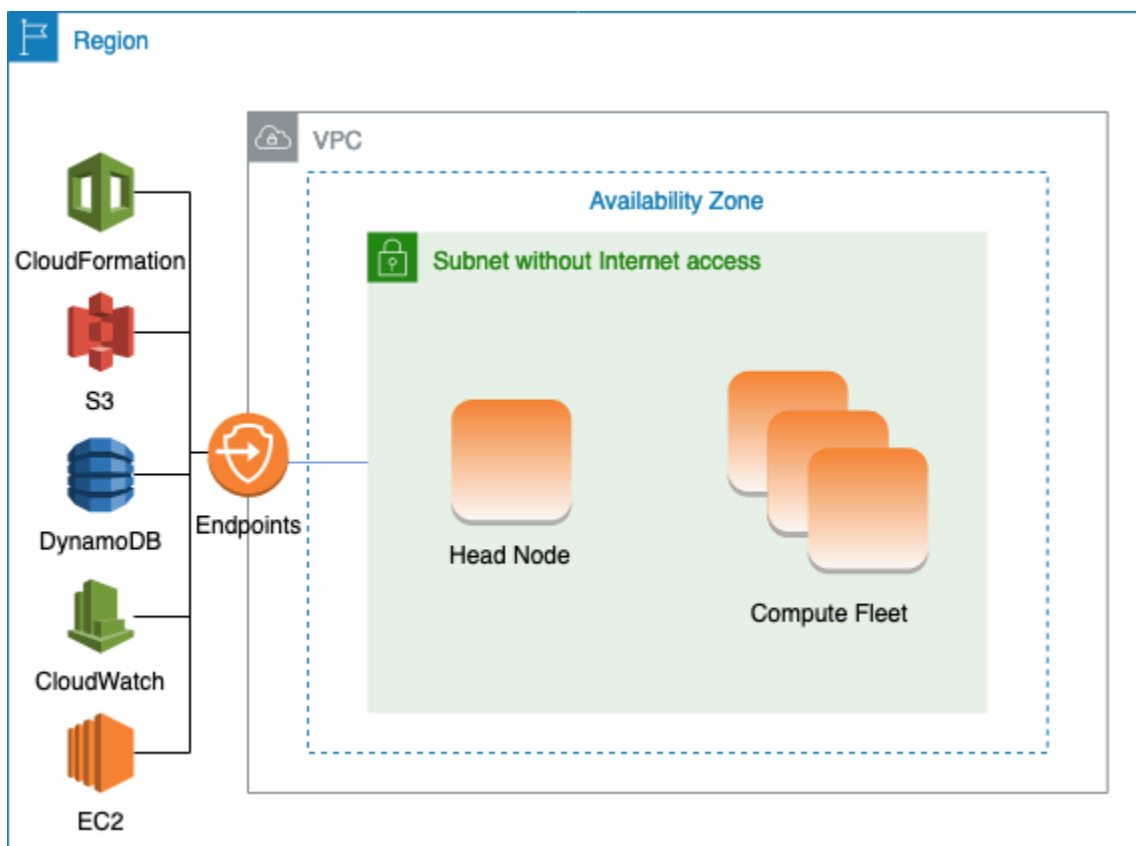
```
KeyName: ec2-key-name
Scheduling:
  Scheduler: awsbatch
  AwsBatchQueues:
    - ...
  Networking:
    SubnetIds:
      - subnet-23456789 # subnet with internet gateway, NAT gateway or proxy
      #AssignPublicIp: true | false
```

Na seção [Scheduling](#) / [AwsBatchQueues](#) / [Networking](#), o [SubnetIds](#) é um tipo de lista, mas, atualmente, somente uma sub-rede é suportada.

Para obter mais informações, consulte os tópicos a seguir:

- [Ambientes de computação gerenciado do AWS Batch](#)
- [Trabalhos em paralelo de vários nós do AWS Batch](#)
- [Redes de tarefas do Amazon ECS com o modo de rede awsvpc](#)

AWS ParallelCluster em uma sub-rede única sem acesso à internet



Uma sub-rede sem acesso à Internet não permite conexões de entrada ou saída com a Internet. Essa configuração do AWS ParallelCluster pode ajudar clientes preocupados com a segurança a aprimorar ainda mais a segurança de seus recursos do AWS ParallelCluster. Os nós do AWS ParallelCluster são criados a partir de AWS ParallelCluster AMIs que incluem todo o software necessário para executar um cluster sem acesso à Internet. Dessa forma, com o AWS ParallelCluster é possível criar e gerenciar clusters com nós que não têm acesso à Internet.

Nesta seção, você aprenderá como configurar o cluster. Você também aprenderá sobre as limitações na execução de clusters sem acesso à Internet.

Configurando endpoints de VPC

Para garantir o funcionamento adequado do cluster, os nós do cluster devem ser capazes de interagir com vários serviços AWS.

Crie e configure os seguintes [VPC endpoints](#) para que os nós do cluster possam interagir com os Serviços AWS, sem acesso à Internet:

Commercial and AWS GovCloud (US) partitions


Serviço	Nome do serviço	Tipo
Amazon CloudWatch	com.amazonaws. <i>region-id</i> .logs	Interface
AWS CloudFormation	com.amazonaws. <i>region-id</i> .cloudformation	Interface
Amazon EC2	com.amazonaws. <i>region-id</i> .ec2	Interface
Amazon S3	com.amazonaws. <i>region-id</i> .s3	Gateway
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	Gateway
AWS Secrets Manager**	com.amazonaws. <i>region-id</i> .secretsmanager	Interface

China partition

Serviço	Nome do serviço	Tipo
Amazon CloudWatch	com.amazonaws. <i>region-id</i> .logs	Interface
AWS CloudFormation	cn.com.amazonaws. <i>region-id</i> .cloudformation	Interface
Amazon EC2	cn.com.amazonaws. <i>region-id</i> .ec2	Interface
Amazon S3	com.amazonaws. <i>region-id</i> .s3	Gateway
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	Gateway
AWS Secrets Manager**	com.amazonaws. <i>region-id</i> .secretsmanager	Interface

** Esse endpoint só é necessário quando [DirectoryService](#) está habilitado, caso contrário, é opcional.

Todas as instâncias na VPC devem ter grupos de segurança adequados para se comunicar com os endpoints. Você pode fazer isso adicionando grupos de segurança a [AdditionalSecurityGroups](#) via [HeadNode](#) e [AdditionalSecurityGroups](#) via configurações de [SlurmQueues](#). Por exemplo, se os endpoints da VPC forem criados sem especificar explicitamente um grupo de segurança, o grupo de segurança padrão será associado aos endpoints. Ao adicionar o grupo de segurança padrão ao [AdditionalSecurityGroups](#), você ativa a comunicação entre o cluster e os endpoints.

 Note

Ao usar políticas do IAM para restringir o acesso aos endpoints da VPC, você deve adicionar o seguinte ao endpoint da VPC do Amazon S3:

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

Desative o Route 53 e use nomes de host EC2

Ao criar um cluster Slurm, o AWS ParallelCluster cria uma zona hospedada privada do Route 53 que é usada para resolver os nomes de host dos nós de computação personalizados, como `{queue_name}-{st|dy}-{compute_resource}-{N}`. Como o Route 53 não é compatível com endpoints da VPC, esse recurso deve ser desativado. Além disso, AWS ParallelCluster deve ser configurado para usar os nomes de host padrão do EC2, como `ip-1-2-3-4`. Aplique as seguintes configurações à configuração do seu cluster:

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

Warning

Para clusters criados com [SlurmSettings](#) / [Dns](#) / [DisableManagedDns](#) e [UseEc2Hostnames](#) definidos como `true`, o `NodeName` do Slurm não é resolvido pelo DNS. Em vez disso, use o `NodeHostName` do Slurm.

Note

Esta nota não é relevante a partir do AWS ParallelCluster versão 3.3.0.

Para versões compatíveis do AWS ParallelCluster anteriores à 3.3.0:

Quando `UseEc2Hostnames` é definido como `true`, o arquivo de configuração do Slurm é definido com os scripts `prolog` e `epilog` do AWS ParallelCluster:

- O `prolog` é executado para adicionar informações sobre nós ao `/etc/hosts` nos nós de computação quando cada trabalho é alocado.
- O `epilog` é executado para limpar o conteúdo escrito pelo `prolog`.

Para adicionar scripts personalizados `prolog` ou `epilog`, adicione-os às pastas `/opt/slurm/etc/pcluster/prolog.d/` ou `/opt/slurm/etc/pcluster/epilog.d/`, respectivamente.

Configuração do cluster

Saiba como configurar seu cluster para ser executado em uma sub-rede sem conexão com a Internet.

A configuração para essa arquitetura requer as seguintes definições:

```
# 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
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\(s\)](#): A sub-rede sem acesso à Internet.

Para permitir a comunicação entre os serviços AWS ParallelCluster e AWS, a VPC da sub-rede deve ter os VPC endpoints conectados. Antes de criar seu cluster, verifique se a [atribuição automática de endereço IPv4 público está desativada](#) na sub-rede para garantir que os comandos `pcluster` tenham acesso ao cluster.

- [AdditionalSecurityGroups](#): o grupo de segurança que permite a comunicação entre o cluster e os endpoints da VPC.

Opcional:

- Se os endpoints da VPC forem criados sem especificar explicitamente um grupo de segurança, o grupo de segurança padrão da VPC será associado. Portanto, forneça o grupo de segurança padrão em `AdditionalSecurityGroups`.
- Se grupos de segurança personalizados forem usados ao criar o cluster e/ou os endpoints da VPC, `AdditionalSecurityGroups` será desnecessário, desde que os grupos de segurança personalizados permitam a comunicação entre o cluster e os endpoints da VPC.
- [Scheduler](#): O programador do cluster.

`slurm` é o único valor válido. Somente o programador Slurm oferece suporte a um cluster em uma sub-rede sem acesso à Internet.

- [SlurmSettings](#): As configurações do Slurm.

Consulte a seção anterior Desativar o Route53 e usar nomes de host EC2.

Limitações

- Conectando-se ao nó principal via SSH ou NICE DCV: Ao se conectar a um cluster, certifique-se de que o cliente da conexão possa acessar o nó principal do cluster por meio de seu endereço IP privado. Se o cliente não estiver na mesma VPC do nó principal, use uma instância de proxy em uma sub-rede pública da VPC. Esse requisito se aplica às conexões SSH e DCV. O IP público de um nó principal não fica acessível se a sub-rede não tiver acesso à Internet. Os comandos

`pcluster ssh` e `dcv-connect` usam o IP público, se existir, ou o IP privado. Antes de criar seu cluster, verifique se a [atribuição automática de endereço IPv4 público está desativada](#) na sub-rede para garantir que os comandos `pcluster` tenham acesso ao cluster.

O exemplo a seguir mostra como você pode se conectar a uma sessão DCV em execução no nó principal do seu cluster. Você se conecta por meio de uma instância proxy do EC2. A instância funciona como um servidor NICE DCV para seu PC e como cliente para o nó principal na sub-rede privada.

Conecte-se por DCV por meio de uma instância de proxy em uma sub-rede pública:

1. Crie uma instância do EC2 em uma sub-rede pública, que esteja na mesma VPC da sub-rede do cluster.
 2. Certifique-se de que o cliente e o servidor do NICE DCV estejam instalados na sua instância do EC2.
 3. Anexe uma política de usuário AWS ParallelCluster à instância de proxy do EC2. Para obter mais informações, consulte [AWS ParallelCluster exemplo de políticas `pcluster` de usuário](#).
 4. Instale AWS ParallelCluster na instância proxy do EC2.
 5. Conecte-se via DCV à instância proxy do EC2.
 6. Use o comando `pcluster dcv-connect` na instância do proxy para se conectar ao cluster contido na sub-rede sem acesso à Internet.
- Interagindo com outros serviços da AWS: Somente os serviços estritamente exigidos pelo AWS ParallelCluster estão listados acima. Se seu cluster precisar interagir com outros serviços, crie os VPC endpoints correspondentes.

Nós de login

A partir da versão 3.7.0, os administradores de cluster do AWS ParallelCluster podem provisionar nós de login que podem ser usados para fornecer acesso aos usuários para executar trabalhos em vez de acessar diretamente o nó principal do cluster. Os usuários do cluster com as permissões apropriadas podem usar o Active Directory ou sua credencial `ssh` para fazer login, enviar e gerenciar seus trabalhos. Como resultado, o gerenciamento do cluster pode ser aprimorado e as chances de esgotar os recursos do nó principal exigidos pelo Slurm para gerenciar o cluster podem ser minimizados. Os usuários conectados também terão acesso a todo o armazenamento compartilhado do cluster montado nos nós de login. Se os nós de login precisarem ser interrompidos, os usuários conectados serão notificados com antecedência por meio da sessão ativa do shell que estão usando.

Os nós de login são especificados como pools em que um pool define um grupo de nós de login que têm a mesma configuração de recursos. Todos os nós de login em um pool são configurados para fazer parte de um [Network Load Balancer](#) que permite a distribuição de sessões entre os nós de login em procedimento round-robin. A implementação atual permite especificar um pool de nós de login contendo vários nós de login.

Segurança

Os nós de login herdam as configurações de AllowedIPs [AllowedIps](#) do nó principal. Dessa forma, os administradores do cluster podem restringir a postura de segurança do cluster especificando o CIDR de origem ou uma lista de prefixos de onde as conexões SSH são permitidas.

Na implementação atual, o acesso ao nó principal não é automaticamente restrito ao ativar os nós de login. Se necessário, um administrador de cluster pode restringir esse acesso atualizando a configuração ssh dos nós principais usando comandos Linux padrão. Isso também pode ser feito especificando grupos de segurança personalizados no nó principal usando a configuração `AdditionalSecurityGroups` na seção do nó principal do arquivo YAML do ParallelCluster para recusar conexões de usuários não autorizados.

Redes

Os nós de login são provisionados com um único endereço de conexão para o Network Load Balancer configurado para o pool de nós de login. As configurações de conectividade do endereço são baseadas no tipo de sub-rede especificado na configuração do Pool de nós de login.

- Se a sub-rede for privada, o endereço será privado e, para conceder acesso aos nós de login, o administrador do cluster deverá provisionar um bastion host.
- Se a sub-rede for pública, o endereço será público

Todas as solicitações de conexão são gerenciadas pelo Network Load Balancer usando roteamento round-robin.

Armazenamento

Todo o armazenamento compartilhado configurado no cluster usando o ParallelCluster, incluindo o armazenamento gerenciado, será montado em todos os nós de login.

Recuperar informações dos nós de login

Para recuperar o endereço da única conexão provisionada para acessar os nós de login, o administrador do cluster pode executar o comando [describe-cluster](#). O comando também fornecerá mais informações sobre o status dos nós de login.

Os nós de login são um novo tipo de nó compatível com o ParallelCluster que pode ser especificado com o comando [describe-cluster-instances](#) ao consultar o status de um tipo de nó específico.

A disponibilidade de um único endereço de conexão para o pool de nós de login não impede o acesso direto a um nó de login específico. No entanto, não é recomendável usar a conexão direta para evitar avisos do cliente ssh. O cliente ssh armazena identificadores de host localmente para cada endereço de destino. Como o identificador do host é específico por pool, o uso de diferentes IPs e/ou do único endereço de conexão pode ter o mesmo identificador de host associado a diferentes endereços de destino: isso pode causar um aviso do cliente ssh, pois o mesmo identificador de host está associado a vários destinos.

Propriedades de Imds

O acesso ao IMDS do nó de login (e às credenciais do perfil de instância) é restrito ao usuário raiz, usuário administrativo do cluster (`pc-cluster-admin` por padrão) e usuário padrão específico do sistema operacional (`ec2-user` no Amazon Linux 2 e RedHat, `ubuntu` no Ubuntu 18.04, `centos` no CentOS 7).

Para restringir o acesso ao IMDS, o AWS ParallelCluster gerencia uma cadeia de `iptables`.

Note

Qualquer personalização de regras `iptables` ou `ip6tables` pode interferir no mecanismo usado para restringir o acesso ao IMDS no nó de login. Consulte também [Imds property setting](#).

Ciclo de vida dos nós de login

Atualmente, não há nenhum comando dedicado para parar e iniciar os nós de login em um pool. Para interromper os nós de login em um pool, o administrador do cluster precisa atualizar a configuração do cluster especificando zero na contagem de nós de login (`Count: 0`) e, em seguida, executar um comando [pcluster.update-cluster-v3](#).

Note

Os usuários conectados são notificados sobre o encerramento da instância específica e sobre o período de carência relacionado. Durante o período de carência, nenhuma nova conexão será permitida, exceto as do [usuário padrão do cluster](#). A mensagem mostrada é personalizável pelo administrador do cluster a partir do nó principal ou de um nó de login que edita o arquivo `/opt/parallelcluster/shared_login_nodes/loginmgtd_config.json`.

Para iniciar o pool de nós de login, o administrador do cluster precisa restaurar o valor Count anterior na configuração do cluster e, em seguida, executar um comando [update-cluster](#).

Permissões necessárias para executar o pool de nós de login

Para gerenciar o pool de nós de login, o administrador do cluster deve ter as seguintes permissões adicionais:

```
- Action:
  - 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
  - elasticloadbalancing:AddTags
  - elasticloadbalancing:CreateLoadBalancer
Resource: '*'
```

Effect: Allow

Ações de bootstrap personalizadas

Se você definir as configurações [HeadNode](#) / [CustomActions](#) / [OnNodeStart](#), o AWS ParallelCluster executará um código arbitrário imediatamente após o início do nó. Se você definir as configurações [HeadNode](#) / [CustomActions](#) / [OnNodeConfigured](#), o AWS ParallelCluster executará o código depois que a configuração do nó for concluída corretamente.

A partir do AWS ParallelCluster versão 3.4.0, o código pode ser executado após a atualização do nó principal, se você definir as configurações [HeadNode](#) / [CustomActions](#) / [OnNodeUpdated](#).

Na maioria dos casos, esse código é armazenado no Amazon Simple Storage Service (Amazon S3) e acessado por meio de uma conexão HTTPS. O código é executado como root e pode estar em qualquer linguagem de script compatível com o sistema operacional do cluster. Muitas vezes, o código está em Bash ou Python.

Note

A partir do AWS ParallelCluster versão 3.7.0, o padrão de configuração [Imds](#) / [ImdsSupport](#) do cluster/ é v2.0.

Ao criar um novo cluster para atualizar para a versão 3.7.0 e versões posteriores, atualize seus scripts de ação de bootstrap personalizados para serem compatíveis com o IMDSv2 ou defina [Imds](#) / [ImdsSupport](#) para v1.0 no arquivo de configuração do cluster.

Warning

Você é responsável por configurar os scripts e argumentos personalizados conforme descrito no [modelo de responsabilidade compartilhada](#). Verifique se seus scripts e argumentos de bootstrap personalizados são de fontes nas quais você confia que possam ter acesso total aos nós do cluster.

⚠ Warning

O AWS ParallelCluster não suporta o uso de variáveis internas fornecidas por meio do arquivo `/etc/parallelcluster/cfnconfig`. Esse arquivo pode ser removido como parte de uma versão futura.

As ações `OnNodeStart` são chamadas antes que qualquer ação de bootstrap de implantação de nó seja iniciada, como configurar o NAT, o Amazon Elastic Block Store (Amazon EBS) ou o programador. As ações de bootstrap `OnNodeStart` podem incluir a modificação do armazenamento, a adição de usuários extras e a adição de pacotes.

ℹ Note

Se você configurar [DirectoryService](#) e um script [HeadNode](#) / [CustomActions](#) / [OnNodeStart](#) para seu cluster, AWS ParallelCluster configura `DirectoryService` e reinicia o `sssd`, antes de executar o script `OnNodeStart`.

As ações `OnNodeConfigured` são chamadas após a conclusão dos processos de bootstrap do nó. As ações `OnNodeConfigured` servem como as últimas ações que ocorrem antes que uma instância seja considerada totalmente configurada e concluída. Algumas ações `OnNodeConfigured` podem incluir a alteração de configurações do programador, a modificação do armazenamento ou a modificação de pacotes. Os argumentos podem ser transmitidos para scripts, especificando-os na configuração.

As ações `OnNodeUpdated` são chamadas depois que a atualização do nó principal é concluída e o programador e o armazenamento compartilhado estão alinhados com as alterações mais recentes na configuração do cluster.

Quando as ações personalizadas `OnNodeStart` ou `OnNodeConfigured` são bem-sucedidas, o sucesso é indicado com o código de saída zero (0). Qualquer outro código de saída indica que o bootstrap da instância falhou.

Quando as ações personalizadas `OnNodeUpdated` são bem-sucedidas, o sucesso é sinalizado com o código de saída zero (0). Qualquer outro código de saída indica que a atualização falhou.

Note

Se você configurar [OnNodeUpdated](#), deverá restaurar manualmente as ações [OnNodeUpdated](#) para o estado anterior em caso de falhas de atualização. Se uma ação personalizada [OnNodeUpdated](#) falhar, a atualização voltará ao estado anterior. No entanto, a ação [OnNodeUpdated](#) só é executada no momento da atualização e não no momento da reversão da pilha.

Você pode especificar scripts diferentes para o nó principal e para cada fila, nas seções de configuração [HeadNode](#) / [CustomActions](#) e [Scheduling](#) / [SlurmQueues](#) / [CustomActions](#). [OnNodeUpdated](#) só pode ser configurado na seção [HeadNode](#).

Note

Antes do AWS ParallelCluster versão 3.0, não era possível especificar scripts diferentes para os nós principais e de computação. Consulte [Passando do AWS ParallelCluster 2.x para 3.x](#).

Tópicos

- [Configuração](#)
- [Argumentos](#)
- [Exemplo de cluster com ações de bootstrap personalizadas](#)
- [Exemplo de atualização de um script de bootstrap personalizado para IMDSv2](#)
- [Exemplo de atualização de uma configuração para IMDSv1](#)

Configuração

As seguintes configurações são usadas para definir [HeadNode](#) / [CustomActions](#) / [OnNodeStart](#) & [OnNodeConfigured](#) & [OnNodeUpdated](#) e ações e argumentos [Scheduling](#) / [CustomActions](#) / [OnNodeStart](#) & [OnNodeConfigured](#).

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
```

```
# Script URL. This is run before any of the bootstrap scripts are run
Script: s3://bucket-name/on-node-start.sh
Args:
  - arg1
OnNodeConfigured:
# Script URL. This is run after all the bootstrap scripts are run
Script: s3://bucket-name/on-node-configured.sh
Args:
  - arg1
OnNodeUpdated:
# Script URL. This is run after the head node update is completed.
Script: s3://bucket-name/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://bucket-name/on-node-start.sh
        Args:
          - arg1
      OnNodeConfigured:
        Script: s3://bucket-name/on-node-configured.sh
        Args:
          - arg1
    Iam:
      S3Access:
        - BucketName: bucket_name
          EnableWriteAccess: false
```

Usando a configuração Sequence (adicionada no AWS ParallelCluster versão 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://bucket-name/on-node-start1.sh
        Args:
          - arg1
      - Script: s3://bucket-name/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://bucket-name/on-node-configured1.sh
        Args:
          - arg1
      - Script: s3://bucket-name/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://bucket-name/on-node-updated1.sh
        Args:
          - arg1
      - Script: s3://bucket-name/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://bucket-name/on-node-start1.sh  
        Args:  
          - arg1  
      - Script: s3://bucket-name/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://bucket-name/on-node-configured1.sh  
        Args:  
          - arg1  
      - Script: s3://bucket-name/on-node-configured2.sh  
        Args:  
          - arg1  
    [...]  
Iam:  
  S3Access:  
    - BucketName: bucket_name  
    EnableWriteAccess: false
```

A configuração Sequence foi adicionada a partir do AWS ParallelCluster versão 3.6.0. Ao especificar Sequence, você pode listar vários scripts para uma ação personalizada. AWS ParallelCluster continua oferecendo suporte à configuração de uma ação personalizada com um único script, sem incluir Sequence.

O AWS ParallelCluster não suporta a inclusão de um único script e Sequence para a mesma ação personalizada. Por exemplo, o AWS ParallelCluster falhar ao especificar a configuração a seguir.

```
[...]  
CustomActions:  
  OnNodeStart:  
    # Script URL. This is run before any of the bootstrap scripts are run  
    Script: s3://bucket-name/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://bucket-name/on-node-start1.sh
  Args:
  - arg1
- Script: s3://bucket-name/on-node-start2.sh
  Args:
  - arg1
[...]
```

Argumentos

Note

No AWS ParallelCluster 2.x, os argumentos \$1 eram reservados para armazenar a URL do script personalizado. Se você quiser reutilizar os scripts de bootstrap personalizados criados para AWS ParallelCluster 2.x com o AWS ParallelCluster 3.x, você precisa adaptá-los considerando a mudança dos argumentos. Consulte [Passando do AWS ParallelCluster 2.x para 3.x](#).

Exemplo de cluster com ações de bootstrap personalizadas

As etapas a seguir criam um script simples a ser executado após a configuração do nó, que instala os pacotes R, curl e wget nos nós do cluster.

1. Crie um script.

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

2. Faça upload do script com as permissões corretas para o Amazon S3. Se as permissões de leitura pública não forem apropriadas para você, use as sessões de configuração [HeadNode](#) /

[Iam](#) / [S3Access](#) e [Scheduling](#) / [SlurmQueues](#). Para obter mais informações, consulte [Como trabalhar com o Amazon S3](#).

```
$ aws s3 cp --acl public-read /path/to/myscript.sh s3://<bucket-name>/myscript.sh
```

Important

Se o script foi editado no Windows, as terminações de linha devem ser alteradas de CRLF para LF antes que seja feito upload do script para o Amazon S3.

3. Atualize a configuração do AWS ParallelCluster para incluir a nova ação OnNodeConfigured.

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

Se o bucket não tiver permissão de leitura pública, use s3 como o protocolo de URL.

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

4. Execute os clusters.

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

5. Verifique a saída.

- Se você adicionou ações personalizadas à configuração HeadNode, faça login no nó principal e verifique o arquivo `cfn-init.log` localizado em `/var/log/cfn-init.log`, executando o seguinte 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 você adicionou ações personalizadas à configuração SlurmQueues, verifique o `cloud-init.log` localizado no `/var/log/cloud-init.log` em um nó de computação. Use CloudWatch para visualizar esses registros.

Você pode visualizar esses dois registros no CloudWatch console da Amazon. Para obter mais informações, consulte [Integração com Amazon CloudWatch Logs](#).

Exemplo de atualização de um script de bootstrap personalizado para IMDSv2

No exemplo a seguir, atualizamos um script de ação de bootstrap personalizado usado com o IMDSv1 para ser usado com o IMDSv2. O script IMDSv1 recupera os metadados do ID da AMI da instância do EC2.

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

A seguir é mostrado o script de ação de bootstrap personalizado com modificação para ser compatível com o 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
```

Para mais informações, consulte [Recuperar metadados de instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Exemplo de atualização de uma configuração para IMDSv1

Veja a seguir um exemplo de uma configuração de cluster que oferece suporte ao IMDSv1 ao usar o AWS ParallelCluster versões 3.7.0 e anteriores.

```
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
```

Como trabalhar com o Amazon S3

Você pode configurar o acesso do AWS ParallelCluster ao Amazon S3 por meio dos parâmetros [HeadNode](#) / [Iam](#) / [S3Access](#) e [Scheduling](#) / [SlurmQueues](#) / [- Name](#) / [Iam](#) / [S3Access](#) na configuração do AWS ParallelCluster.

Exemplos

O exemplo a seguir configura o acesso somente leitura a todos os objetos em *firstbucket/read_only/* e o acesso de leitura/gravação a todos os objetos em *secondbucket/read_and_write/*.

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

O próximo exemplo configura o acesso somente para leitura a todos os objetos na pasta *read_only/* em qualquer bucket (*) na conta.

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

O exemplo final configura o acesso *read_only* a todos os buckets e objetos na conta.

```
...
```

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

Trabalho com Instâncias spot

AWS ParallelCluster usa Instâncias Spot se você tiver definido [SlurmQueues/CapacityType](#) ou [AwsBatchQueues/CapacityTypeSP0T](#) no arquivo de configuração do cluster. As instâncias spot são mais econômicas do que as instâncias sob demanda, mas podem ser interrompidas. Isso pode ajudar a aproveitar os avisos de interrupção de instâncias spot, que enviam um aviso de dois minutos antes que o Amazon EC2 interrompa ou encerre a instância spot. Para obter mais informações, consulte [Interrupção de instâncias spot](#) no Manual do usuário do Amazon EC2 para instâncias do Linux. Para saber como [AwsBatchQueues](#) funciona com instâncias spot, consulte [Recursos de computação](#) no Guia do usuário AWS Batch.

O programador AWS ParallelCluster configurado atribui trabalhos a recursos computacionais em filas com instâncias spot da mesma forma que atribui trabalhos a recursos de computação em filas com instâncias sob demanda.

Ao usar instâncias Spot, uma função vinculada ao serviço `AWSServiceRoleForEC2Spot` deve existir na sua conta. Para criar essa função na sua conta usando a AWS CLI, execute o seguinte comando:

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

Para obter mais informações, consulte [Função vinculada ao serviço para solicitações de instâncias spot](#) no Guia do usuário do Amazon EC2 para instâncias do Linux

As seções a seguir descrevem três cenários nos quais as instâncias spot podem ser interrompidas ao usar [SlurmQueues](#).

Cenário 1: Uma instância spot sem trabalhos em execução é interrompida

Quando ocorrer essa interrupção, o AWS ParallelCluster tentará substituir a instância se a fila do programador tiver trabalhos pendentes que exigem instâncias adicionais ou se o número de instâncias ativas for menor que [SlurmQueues](#) / [ComputeResources](#) / [MinCount](#). Se o AWS

ParallelCluster não conseguir provisionar novas instâncias, uma solicitação de novas instâncias será repetida periodicamente.

Cenário 2: Uma instância spot que executa trabalhos de nó único é interrompida

O trabalho falha com um código de estado de `deNODE_FAIL`, e o trabalho é colocado novamente na fila (a menos que seja `--no-requeue` especificado quando o trabalho é enviado). Se o nó for estático, ele será substituído. Se o nó for um nó dinâmico, o nó será encerrado e redefinido. Para obter mais informações sobre `sbatch`, incluindo o `--no-requeue` parâmetro, consulte [sbatch](#) documentação do Slurm.

Cenário 3: Uma instância spot que executa trabalhos de vários nós é interrompida

O trabalho falha com um código de estado de `deNODE_FAIL`, e o trabalho é colocado novamente na fila (a menos que tenha `--no-requeue` sido especificado quando o trabalho foi enviado). Se o nó for estático, ele será substituído. Se o nó for um nó dinâmico, o nó será encerrado e redefinido. Outros nós que estavam executando os trabalhos encerrados podem ser alocados a outros trabalhos pendentes, ou reduzidos depois que o tempo configurado [SlurmSettings](#) / [ScaledownIdleTime](#) tiver passado.

Para obter mais informações sobre instâncias spot, consulte [Instâncias spot](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Agendadores suportados por AWS ParallelCluster

Agendadores suportados por AWS ParallelCluster

AWS ParallelCluster suporta Slurm e AWS Batch agendadores, definidos usando a [Scheduler](#) configuração.

Tópicos

- [Slurm Workload Manager \(slurm\)](#)
- [AWS Batch \(awsbatch\)](#)

Slurm Workload Manager (**slurm**)

Tamanho e atualização da capacidade do cluster

A capacidade do cluster é definida pelo número de nós de computação que o cluster pode escalar. Os nós de computação são apoiados por instâncias do EC2 definidas nos recursos computacionais da AWS ParallelCluster configuração (`Scheduling/SlurmQueues/ComputeResources`) e são organizados em filas (`Scheduling/SlurmQueues`) que mapeiam 1:1 para partições. Slurm

Em um recurso de computação, é possível configurar o número mínimo de nós de computação (instâncias) que sempre devem ser mantidos em execução no cluster (MinCount) e o número máximo de instâncias para as quais o recurso computacional pode ser escalado (MaxCount3).

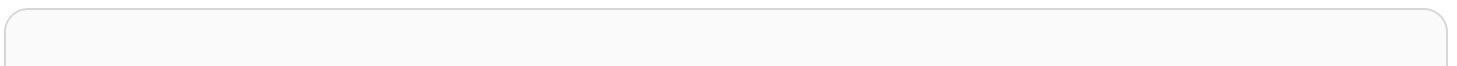
No momento da criação do cluster, ou após uma atualização do cluster AWS ParallelCluster , executa quantas instâncias do EC2 forem configuradas `MinCount` para cada recurso computacional (`Scheduling/SlurmQueues/ ComputeResources`) definido no cluster. As instâncias lançadas para cobrir a quantidade mínima de nós para os recursos computacionais no cluster são chamadas de nós estáticos. Depois de iniciados, os nós estáticos devem ser persistentes no cluster e não são encerrados pelo sistema, a menos que ocorra um evento ou condição específica. Esses eventos incluem, por exemplo, a falha das verificações de integridade do EC2 e a alteração do status do nó Slurm para `DRAIN` Slurm ou `DOWN`.

As instâncias do EC2, na faixa de **1** até '**MaxCount - MinCount**' (**MaxCount** menos **MinCount**), lançadas sob demanda para lidar com o aumento da carga do cluster, são chamadas de nós dinâmicos. Sua natureza é efêmera, eles são iniciados para atender a trabalhos pendentes e são encerrados quando permanecem ociosos por um período de tempo definido `Scheduling/SlurmSettings/ScaledownIdleTime` na configuração do cluster (padrão: 10 minutos).

Os nós estáticos e os nós dinâmicos estão em conformidade com o seguinte esquema de nomenclatura:

- Nódulos estáticos `<Queue/Name>-st-<ComputeResource/Name>-<num>` onde `<num>` = `1..ComputeResource/MinCount`
- Nódulos dinâmicos `<Queue/Name>-dy-<ComputeResource/Name>-<num>` onde `<num>` = `1..(ComputeResource/MaxCount - ComputeResource/MinCount)`

Por exemplo, dada a seguinte AWS ParallelCluster configuração:




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

Os seguintes nós serão definidos em 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 um recurso computacional o tiver `MinCount == MaxCount`, todos os nós de computação correspondentes serão estáticos e todas as instâncias serão iniciadas no momento da criação/atualização do cluster e mantidas em funcionamento. Por exemplo: .

```
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]
```

Atualização da capacidade do cluster

A atualização da capacidade do cluster inclui a adição ou remoção de filas, recursos computacionais ou a alteração MinCount/MaxCount de um recurso computacional. A partir da AWS ParallelCluster versão 3.9.0, reduzir o tamanho de uma fila exige que a frota de computação seja interrompida ou [QueueUpdateStrategy](#) configurada como TERMINATE antes que uma atualização do cluster ocorra. Não é necessário interromper a frota de computação ou configurar [QueueUpdateStrategy](#) para TERMINATE quando:

- Adicionando novas filas ao Scheduling/ [SlurmQueues](#)
- Adicionar novos recursos computacionais Scheduling/SlurmQueues/[ComputeResources](#) a uma fila
- Aumentando a [MaxCount](#) quantidade de um recurso computacional
- Aumento MinCount de um recurso computacional e aumento MaxCount do mesmo recurso computacional em pelo menos a mesma quantidade

Considerações e limitações

Esta seção tem como objetivo descrever quaisquer fatores, restrições ou limitações importantes que devem ser levados em consideração ao redimensionar a capacidade do cluster.

- Ao remover uma fila de Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues todos os nós de computação com nome <Queue/Name> - *, tanto estático quanto dinâmico, ela será removida da Slurm configuração e as instâncias EC2 correspondentes serão encerradas.
- Ao remover um recurso computacional Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues-ComputeResources> ComputeResources de uma fila, todos os nós de computação com nome <Queue/Name> - * - <ComputeResource/Name> - *, estático e dinâmico, serão removidos da Slurm configuração e as instâncias EC2 correspondentes serão encerradas.

Ao alterar o `MinCount` parâmetro de um recurso computacional, podemos distinguir dois cenários diferentes: se `MaxCount` é mantido igual a `MinCount` (somente capacidade estática) e se `MaxCount` é maior que `MinCount` (capacidade estática e dinâmica mista).

Alterações de capacidade somente com nós estáticos

- Se `MinCount == MaxCount`, ao aumentar `MinCount` (e `MaxCount`), o cluster for configurado estendendo o número de nós estáticos até o novo valor de, `MinCount <Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` e o sistema continuar tentando iniciar instâncias do EC2 para atender à nova capacidade estática necessária.
- Se `MinCount == MaxCount`, ao diminuir `MinCount` (e `MaxCount`) da quantidade `N`, o cluster for configurado removendo os últimos `N` nós estáticos `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`] e o sistema encerrará as instâncias EC2 correspondentes.

- Estado inicial `MinCount = MaxCount = 100`

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

- Atualizar `-30` em `MinCount` e `MaxCount`: `MinCount = MaxCount = 70`

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

Alterações de capacidade com nós mistos

Se `MinCount < MaxCount`, `MinCount` ao aumentar em uma quantidade `N` (supondo que `MaxCount` será mantido inalterado), o cluster for configurado estendendo o número de nós estáticos até o novo valor de `MinCount` (`old_MinCount + N`): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` e o sistema continuará tentando iniciar instâncias do EC2 para atender à nova capacidade estática necessária. Além disso, para honrar a `MaxCount` capacidade

do recurso computacional, a configuração do cluster é atualizada removendo os últimos N nós dinâmicos: `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` e o sistema encerrará as instâncias EC2 correspondentes.

- Estado inicial: `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]
```

- Atualize +30 para `MinCount` : `MinCount = 130` (`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 `MinCount < MaxCount`, ao aumentar `MinCount` e `MaxCount` com a mesma quantidade N, o cluster for configurado estendendo o número de nós estáticos para o novo valor de `MinCount` (`old_MinCount + N`): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` e o sistema continuará tentando iniciar instâncias do EC2 para atender à nova capacidade estática necessária. Além disso, nenhuma alteração será feita no número de nós dinâmicos para honrar o novo

`MaxCount` value.

- Estado inicial: `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]
```

- Atualize +30 para 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 $\text{MinCount} < \text{MaxCount}$, ao diminuir a quantidade N (MinCount supondo que MaxCount será mantida inalterada), o cluster será configurado removendo os últimos N nós estáticos dos nós estáticos `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>` e o sistema encerrará as instâncias EC2 correspondentes. Além disso, para honrar a MaxCount capacidade do recurso computacional, a configuração do cluster é atualizada ampliando o número de nós dinâmicos para preencher a lacuna. $\text{MaxCount} - \text{new_MinCount}$: `<Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]` Nesse caso, como esses são nós dinâmicos, nenhuma nova instância do EC2 será iniciada, a menos que o agendador tenha trabalhos pendentes nos novos nós.

- Estado inicial: 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]
```

- Atualização -30 ativada 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 $\text{MinCount} < \text{MaxCount}$, ao diminuir MinCount e MaxCount com a mesma quantidade N , o cluster for configurado removendo os últimos N nós estáticos `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] e o sistema encerrará as instâncias EC2 correspondentes.

Além disso, nenhuma alteração será feita no número de nós dinâmicos para honrar o novo MaxCount valor.

- Estado inicial: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

```
$ 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]
```

- Atualização -30 ativada 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 $\text{MinCount} < \text{MaxCount}$, ao diminuir a quantidade N (MaxCount supondo que MinCount será mantida inalterada), o cluster for configurado removendo os últimos N nós dinâmicos `<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>`] e o sistema encerrará as instâncias EC2 correspondentes no caso de elas estarem em execução. Nenhum impacto é esperado nos nós estáticos.

- Estado inicial: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

```
$ 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]
```

- Atualização -30 ativada 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]
```

Impactos nos empregos

Em todos os casos em que os nós são removidos e as instâncias do EC2 encerradas, um trabalho em lote executado nos nós removidos será colocado novamente na fila, a menos que não haja outros nós que satisfaçam os requisitos do trabalho. Nesse último caso, o trabalho falhará com o status `NODE_FAIL` e desaparecerá da fila; nesse caso, será necessário reenviá-lo manualmente.

Se você planeja realizar uma atualização de redimensionamento do cluster, pode impedir que os trabalhos sejam executados nos nós que serão removidos durante a atualização planejada. Isso é possível configurando os nós a serem removidos na manutenção. Esteja ciente de que a configuração de um nó em manutenção não afetaria as tarefas que eventualmente já estão sendo executadas no nó.

Suponha que, com a atualização planejada de redimensionamento do cluster, você remova o nó `queue-st-computeresource-[9-10]`. Você pode criar uma Slurm reserva com o seguinte comando

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queue-st-
computeresource-[9-10]
```

Isso criará uma Slurm reserva nomeada `maint_for_update` nos nós `queue-st-computeresource-[9-10]`. A partir do momento em que a reserva é criada, nenhuma outra tarefa pode ser executada nos nós `queue-st-computeresource-[9-10]`. Esteja ciente de que a reserva não impedirá que os trabalhos sejam eventualmente alocados nos nós `queue-st-computeresource-[9-10]`.

Após a atualização do redimensionamento do cluster, se a Slurm reserva tiver sido definida somente nos nós que foram removidos durante a atualização do redimensionamento, a reserva de manutenção será excluída automaticamente. Se, em vez disso, você tiver criado uma Slurm reserva nos nós que ainda estão presentes após a atualização do redimensionamento do cluster, talvez queiramos remover a reserva de manutenção nos nós após a execução da atualização de redimensionamento, usando o seguinte comando

```
sudo -i scontrol delete ReservationName=maint_for_update
```

Para obter detalhes adicionais sobre a Slurm reserva, consulte o documento oficial do SchedMD [aqui](#).

Processo de atualização do cluster sobre mudanças de capacidade

Após uma alteração na configuração do agendador, as seguintes etapas são executadas durante o processo de atualização do cluster:

- Pare AWS ParallelCluster clustermgtd (`supervisorctl stop clustermgtd`)
- Gere a configuração de Slurm partições atualizada a partir da AWS ParallelCluster configuração
- Reiniciar `slurmctld` (feito por meio da receita do serviço Chef)
- Verifique o `slurmctld` status (`systemctl is-active --quiet slurmctld.service`)
- Recarregar configuração Slurm (`scontrol reconfigure`)
- Iniciar `clustermgtd` (`supervisorctl start clustermgtd`)

Para obter informações sobre Slurm, consulte <https://slurm.schedmd.com>. Para downloads, consulte <https://github.com/SchedMD/slurm/tags>. Para obter o código-fonte, consulte <https://github.com/SchedMD/slurm>.

AWS ParallelCluster versão (ões)	Versão do Slurm compatível
3.9.0	23.11.4
3.8.0	23.02.7
3.7.2	23.02.6
3.7.1	23.02.5

AWS ParallelCluster versão (ões)	Versão do Slurm compatível
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

Tópicos

- [Configuração de várias filas](#)
- [Guia do Slurm para o modo de várias filas](#)
- [Modo protegido por cluster do Slurm](#)
- [Failover rápido de capacidade insuficiente do cluster Slurm](#)
- [Programação baseada em memória do Slurm](#)
- [Alocação a vários tipos de instância com o Slurm](#)
- [Dimensionamento de clusters para nós dinâmicos](#)
- [Slurmcontabilidade com AWS ParallelCluster](#)
- [Personalização de configuração do Slurm](#)
- [Slurm prolog e epilog](#)
- [Tamanho e atualização da capacidade do cluster](#)

Configuração de várias filas

Configuração de várias filas

Com o AWS ParallelCluster versão 3, você pode configurar várias filas definindo o [Scheduler](#) para `slurm` e especificando mais de uma fila para [SlurmQueues](#) no arquivo de configuração. Nesse modo, diferentes tipos de instância coexistem nos nós de computação especificados na seção [ComputeResources](#) do arquivo de configuração. [ComputeResources](#) com diferentes tipos de instância têm a escala aumentada ou reduzida conforme necessário para o [SlurmQueues](#).

Filas de clusters e cotas de recursos de computação

Recurso	Cota
Slurm queues	50 filas por cluster
Compute resources	50 recursos de computação por fila 50 recursos de computação por cluster

Contagem de nós

Cada recurso de computação em [ComputeResources](#) para uma fila deve ter um [Name](#), [InstanceType](#), [MinCount](#), e [MaxCount](#) exclusivos. [MinCount](#) e [MaxCount](#) têm valores padrão que definem o intervalo de instâncias de um recurso de computação em [ComputeResources](#) para uma fila. Você também pode especificar seus próprios valores para [MinCount](#) e [MaxCount](#). Cada recurso de computação em [ComputeResources](#) é composto por nós estáticos numerados de 1 até o valor de [MinCount](#) e nós dinâmicos numerados do valor de [MinCount](#) até o valor de [MaxCount](#).

Exemplo de configuração

A seguir, veja um exemplo de uma seção de [Programação](#) para um arquivo de configuração de cluster. Nessa configuração, há duas filas nomeadas `queue1` e `queue2` e cada uma delas tem uma [ComputeResources](#) com uma [MaxCount](#) especificada.

```
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
```

Hostnames

As instâncias que são lançadas na frota de computação são atribuídas dinamicamente. Os nomes de host são gerados para cada nó. Por padrão, AWS ParallelCluster usará o seguinte formato do nome do host:

```
$HOSTNAME=$QUEUE-$STATDYN-$COMPUTE_RESOURCE-$NODENUM
```

- \$QUEUE é o nome da fila. Por exemplo, se a seção [SlurmQueues](#) tiver uma entrada com o [Name](#) definido como “queue-name”, então “\$QUEUE” será “queue-name”.
- \$STATDYN é st para nós estáticos ou dy para nós dinâmicos.
- \$COMPUTE_RESOURCE é o [Name](#) do recurso de computação [ComputeResources](#) correspondente a esse nó.
- \$NODENUM é o número do nó. \$NODENUM fica entre um (1) e o valor de [MinCount](#) para nós estáticos e entre um (1) e [MaxCount-MinCount](#) para nós dinâmicos.

Do arquivo de configuração de exemplo acima, um determinado nó da queue1 e um recurso de computação c5xlarge têm um nome de host: queue1-dy-c5xlarge-1.

Tanto os nomes de host quanto os nomes de domínio totalmente qualificados (FQDN) são criados usando zonas hospedadas do Amazon Route 53. O FQDN é \$HOSTNAME.\$CLUSTERNAME.pcluster, sendo \$CLUSTERNAME o nome do cluster.

Observe que o mesmo formato também será usado para os nomes dos nós do Slurm.

Os usuários podem escolher usar o nome de host EC2 padrão da instância que alimenta o nó de computação em vez do formato de nome de host padrão usado por AWS ParallelCluster. Isso pode ser feito definindo o [UseEc2Hostnames](#) parâmetro como verdadeiro. No entanto, os nomes dos nós do Slurm continuarão usando o formato padrão AWS ParallelCluster.

Guia do Slurm para o modo de várias filas

Aqui você pode aprender como AWS ParallelCluster Slurm gerenciar nós de fila (partição) e como monitorar os estados da fila e do nó.

Visão geral

A arquitetura de escalabilidade é baseada no [Cloud Scheduling Guide](#) do Slurm e no plug-in de economia de energia. Para obter mais informações sobre o plug-in de economia de energia, consulte o [Guia de economia de energia do Slurm](#). Na arquitetura, os recursos com potencial para serem disponibilizados em um cluster geralmente são predefinidos na configuração do Slurm como nós de nuvem.

Ciclo de vida do nó da nuvem

Durante todo o ciclo de vida, os nós da nuvem entram em vários, se não em todos, dos seguintes estados: `POWER_SAVING`, `POWER_UP` (`pow_up`), `ALLOCATED` (`alloc`) e `POWER_DOWN` (`pow_dn`). Em alguns casos, um nó da nuvem pode entrar no estado `OFFLINE`. A lista a seguir detalha vários aspectos desses estados no ciclo de vida do nó na nuvem.

- Um nó em um estado **POWER_SAVING** aparece com um sufixo `~` (por exemplo, `idle~`) em `sinfo`. Nesse estado, nenhuma instância do EC2 serve de apoio ao nó. No entanto, Slurm ainda pode alocar trabalhos para o nó.
- Um nó em transição para um estado **POWER_UP** aparece com um sufixo `#` (por exemplo, `idle#`) em `sinfo`. Um nó faz a transição automática para um estado `POWER_UP` quando Slurm aloca uma tarefa para um nó em um estado `POWER_SAVING`.

Como alternativa, você pode fazer manualmente a transição dos nós para o estado `POWER_UP` como usuário raiz `su` com o comando:

```
$ scontrol update nodename=nodename state=power_up
```

Nesse estágio, o `ResumeProgram` é invocado, as instâncias do EC2 são iniciadas e configuradas, e ocorre a transição do nó para o estado `POWER_UP`.

- Um nó atualmente disponível para uso aparece sem um sufixo (por exemplo `idle`) em `sinfo`. Depois que o nó é configurado e se une ao cluster, ele fica disponível para executar trabalhos. Nesse estágio, o nó está configurado corretamente e pronto para uso.

Como regra geral, recomendamos que o número de instâncias do EC2 seja igual ao número de nós disponíveis. Na maioria dos casos, os nós estáticos ficam disponíveis após a criação do cluster.

- Um nó em transição para um estado **POWER_DOWN** aparece com um sufixo % (por exemplo, idle %) em `sinfo`. Os nós dinâmicos entram automaticamente no estado **POWER_DOWN** depois do [ScaledownIdleTime](#). Por outro lado, os nós estáticos na maioria dos casos não são desligados. No entanto, você pode colocar os nós no estado **POWER_DOWN** manualmente como usuário raiz su com o comando:

```
$ scontrol update nodename=nodename state=down reason="manual draining"
```

Nesse estado, as instâncias associadas a um nó são encerradas e o nó volta ao estado **POWER_SAVING** e fica disponível para uso após o [ScaledownIdleTime](#).

A opção [ScaledownIdleTime](#) é salva na opção `SuspendTimeout` da configuração Slurm.

- Um nó que está off-line aparece com um sufixo * (por exemplo, down*) em `sinfo`. Um nó fica off-line se o controlador Slurm não conseguir entrar em contato com o nó ou se os nós estáticos forem desativados e as instâncias de backup forem encerradas.

Considere os estados dos nós mostrados no exemplo `sinfo` a seguir.

```
$ 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]
```

Os nós `spot-st-spotcompute2-[1-2]` e `efa-st-efacompute1-1` já têm instâncias de backup configuradas e estão disponíveis para uso. Os nós `ondemand-dy-ondemandcompute1-[1-2]` estão no estado **POWER_UP** e devem estar disponíveis em instantes. O nó `gpu-dy-`

gpucompute1-1 está no estado `POWER_DOWN` e passa para o estado `POWER_SAVING` depois de [ScaledownIdleTime](#) (o padrão é 10 minutos).

Todos os outros nós estão no estado `POWER_SAVING` sem nenhuma instância do EC2 como backup.

Trabalhando com um nó disponível

Um nó disponível é apoiado por uma instância do EC2. Por padrão, o nome do nó pode ser usado para fazer SSH diretamente na instância (por exemplo, `ssh efa-st-efacompute1-1`). O endereço IP privado da instância pode ser recuperado usando o comando:

```
$ scontrol show nodes nodename
```

Verifique o endereço IP exibido no campo `NodeAddr`.

Para os nós que não estão disponíveis, o campo `NodeAddr` não deve apontar para uma instância do EC2 em execução. Em vez disso, deve ser igual ao nome do nó.

Estados do trabalho e envio

Na maioria dos casos, os trabalhos enviados são imediatamente alocados aos nós do sistema ou colocados como pendentes se todos os nós estiverem alocados.

Se os nós alocados para um trabalho incluírem qualquer nó em um estado `POWER_SAVING`, o trabalho começará com um estado `CF` ou `CONFIGURING`. Nesse momento, o trabalho aguarda para que os nós no estado `POWER_SAVING` façam a transição para o estado `POWER_UP` e fiquem disponíveis.

Depois que todos os nós alocados para um trabalho estiverem disponíveis, o trabalho entrará no estado `RUNNING (R)`.

Por padrão, todos os trabalhos são enviados para a fila padrão (conhecida como partição em Slurm). Isso é representado por um sufixo `*` após o nome da fila. Você pode selecionar uma fila usando a opção de envio de trabalho `-p`.

Todos os nós são configurados com os seguintes recursos, que podem ser usados nos comandos de envio de tarefas:

- Um tipo de instância (por exemplo, `c5.xlarge`)
- Um tipo de nó (que pode ser `dynamic` ou `static`.)

Você pode ver os recursos de um determinado nó usando o comando:

```
$ scontrol show nodes nodename
```

Quando retornar, confira a lista AvailableFeatures.

Considere o estado inicial do cluster, que você pode ver executando o comando 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   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]
```

Observe que a lista padrão é spot. É indicada pelo sufixo *.

Envia um trabalho para um nó estático na fila padrão (spot).

```
$ sbatch --wrap "sleep 300" -N 1 -C static
```

Envia um trabalho para um nó dinâmico na fila EFA.

```
$ sbatch --wrap "sleep 300" -p efa -C dynamic
```

Envia um trabalho para oito (8) nós c5.2xlarge e dois (2) nós t2.xlarge na fila ondemand.

```
$ sbatch --wrap "sleep 300" -p ondemand -N 10 -C "[c5.2xlarge*8&t2.xlarge*2]"
```

Envia um trabalho para um nó da GPU na fila gpu.

```
$ sbatch --wrap "sleep 300" -p gpu -G 1
```

Considere o estado dos trabalhos usando o comando squeue.

```
$ 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

Os trabalhos 7 e 8 (nas filas spot e efa) já estão em execução (R). Os trabalhos 12 e 13 ainda estão com configuração em andamento (CF), provavelmente aguardando a disponibilização das instâncias.

```
# 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
```

Recursos e estado do nó

Na maioria dos casos, os estados dos nós são totalmente gerenciados de AWS ParallelCluster acordo com os processos específicos no ciclo de vida do nó na nuvem descritos anteriormente neste tópico.

No entanto, AWS ParallelCluster também substitui ou encerra nós não íntegros em DRAINED estados DOWN e nós que têm instâncias de backup não íntegras. Para ter mais informações, consulte [clustermgtd](#).

Estados de partição

AWS ParallelCluster suporta os seguintes estados de partição. Uma partição Slurm é uma fila no AWS ParallelCluster.

- UP: indica que a partição está em um estado ativo. Esse é o valor padrão de uma partição. Nesse estado, todos os nós da partição ficam ativos e disponíveis para uso.
- INACTIVE: indica que a partição está em um estado inativo. Nesse estado, todas as instâncias de backup dos nós de uma partição inativa são encerradas. Novas instâncias não são iniciadas para nós em uma partição inativa.

cluster update-compute-fleet

- Interrompendo a frota de computação - Quando o comando a seguir é executado, todas as partições passam para o INACTIVE estado e AWS ParallelCluster os processos mantêm as partições no estado. INACTIVE

```
$ pcluster update-compute-fleet --cluster-name testSlurm \  
--region eu-west-1 --status STOP_REQUESTED
```

- Iniciando a frota de computação - Quando o comando a seguir é executado, todas as partições inicialmente passam para o estado UP. No entanto, AWS ParallelCluster os processos não mantêm a partição em um UP estado. Você precisa alterar os estados das partições manualmente. Todos os nós estáticos ficam disponíveis após alguns minutos. Observe que definir uma partição como UP não ativa nenhuma capacidade dinâmica.

```
$ pcluster update-compute-fleet --cluster-name testSlurm \  
--region eu-west-1 --status START_REQUESTED
```

Quando update-compute-fleet executado, você pode verificar o estado do cluster executando o comando `pcluster describe-compute-fleet` e verificando Status. A seguir, listamos estados possíveis:

- STOP_REQUESTED: a solicitação de interrupção da frota de computação é enviada ao cluster.
- STOPPING: no momento, o processo `pcluster` está interrompendo a frota de computação.
- STOPPED: o processo `pcluster` finalizou o processo de interrupção, todas as partições estão em estado INACTIVE e todas as instâncias de computação foram encerradas.
- START_REQUESTED: a solicitação de inicialização da frota de computação é enviada ao cluster.
- STARTING: o processo `pcluster` está iniciando o cluster no momento.
- RUNNING: o processo `pcluster` finalizou o processo inicial, todas as partições estão no estado UP e os nós estáticos ficam disponíveis após alguns minutos.

- **PROTECTED**: esse status indica que algumas partições têm falhas de bootstrap consistentes. As partições afetadas estão inativas. Investigue o problema e, em seguida, execute `update-fleet` para reativar a frota.

Controle manual de filas

Em alguns casos, talvez você queira ter algum controle manual sobre os nós ou sobre a fila (conhecida como partição em Slurm) em um cluster. Você pode gerenciar nós em um cluster por meio dos seguintes procedimentos comuns usando o comando `scontrol`.

- Ative os nós dinâmicos no estado **POWER_SAVING**

Execute o comando como usuário raiz `su`:

```
$ scontrol update nodename=nodename state=power_up
```

Você também pode enviar um trabalho `sleep 1` como espaço reservado, solicitando um determinado número de nós e, em seguida, contar com o Slurm para ativação do número necessário de nós.

- Desligue os nós dinâmicos antes de [ScaledownIdleTime](#)

Recomendamos que você defina nós dinâmicos para DOWN como usuário raiz `su` com o comando:

```
$ scontrol update nodename=nodename state=down reason="manually draining"
```

AWS ParallelCluster encerra e redefine automaticamente os nós dinâmicos abatidos.

Em geral, não recomendamos definir os nós para `POWER_DOWN` diretamente com o comando `scontrol update nodename=nodename state=power_down`. Isso porque AWS ParallelCluster gerencia automaticamente o processo de desligamento.

- Desativar uma fila (partição) ou interromper todos os nós estáticos em uma partição específica

Defina uma fila específica para `INACTIVE` como usuário raiz `su` com o comando:

```
$ scontrol update partition=queuename state=inactive
```

Isso encerra todas as instâncias de backup de nós na partição.

- Ativar uma fila (partição)

Defina uma fila específica para UP como usuário raiz su com o comando:

```
$ scontrol update partition=queuename state=up
```

Comportamento e ajustes de escalabilidade

Veja a seguir o exemplo do fluxo de trabalho de escalabilidade normal:

- O programador recebe um trabalho que requer dois nós.
- O programador faz a transição de dois nós para um estado POWER_UP e chama ResumeProgram com os nomes dos nós (por exemplo, queue1-dy-spotcompute1-[1-2]).
- ResumeProgram inicia duas instâncias do EC2 e atribui os endereços IP e nomes de host privados de queue1-dy-spotcompute1-[1-2], aguardando ResumeTimeout (o período padrão é de 30 minutos) antes de redefinir os nós.
- As instâncias são configuradas e se unem ao cluster. Um trabalho começa a ser executado nas instâncias.
- O trabalho é concluído e para de ser executado.
- Depois de decorrido o SuspendTime configurado (que está definido como [ScaledownIdleTime](#)), o programador define as instâncias para o estado POWER_SAVING. Em seguida, o programador define queue1-dy-spotcompute1-[1-2] para o estado POWER_DOWN e chama SuspendProgram com os nomes dos nós.
- SuspendProgram é chamado para dois nós. Os nós permanecem no estado POWER_DOWN, por exemplo, permanecendo idle% por um SuspendTimeout (o período padrão é 120 segundos (2 minutos)). Depois de clustermgtd detectar que os nós estão sendo desligados, ele encerra as instâncias de backup. Em seguida, ele passa queue1-dy-spotcompute1-[1-2] para o estado ocioso e redefine o endereço IP privado e o nome do host para que esteja pronto para ser ativado em trabalhos futuros.

Se algo der errado e uma instância de um determinado nó não puder ser iniciada por algum motivo, acontece o seguinte:

- O programador recebe um trabalho que requer dois nós.

- O programador faz a transição de dois nós de expansão na nuvem para o estado `POWER_UP` e chama `ResumeProgram` com os nomes dos nós (por exemplo `queue1-dy-spotcompute1-[1-2]`).
- `ResumeProgram` inicia somente uma (1) instância do EC2 e configura `queue1-dy-spotcompute1-1`, com uma (1) instância, `queue1-dy-spotcompute1-2`, falhando na inicialização.
- `queue1-dy-spotcompute1-1` não é afetado e fica on-line depois de chegar ao estado `POWER_UP`.
- `queue1-dy-spotcompute1-2` faz a transição para o estado `POWER_DOWN` e o trabalho é enfileirado automaticamente porque Slurm detecta uma falha no nó.
- `queue1-dy-spotcompute1-2` fica disponível após `SuspendTimeout` (o padrão é 120 segundos (2 minutos)). Enquanto isso, o trabalho é recolocado na fila e pode começar a ser executado em outro nó.
- O processo acima se repete até que o trabalho possa ser executado em um nó disponível sem que ocorra uma falha.

Há dois parâmetros de temporização que podem ser ajustados, se necessário:

- **ResumeTimeout** (o padrão é 30 minutos): `ResumeTimeout` controla o tempo que Slurm espera antes da transição do nó para o estado inativo.
 - Pode ser útil estender `ResumeTimeout` se o processo de pré/pós-instalação demorar tanto assim.
 - `ResumeTimeout` também é o tempo máximo que AWS ParallelCluster espera antes de substituir ou redefinir um nó se houver algum problema. Os nós de computação terminam automaticamente se ocorrer algum erro durante a inicialização ou a configuração. AWS ParallelCluster os processos substituem um nó após a detecção de uma instância encerrada.
- **SuspendTimeout** (o padrão é 120 segundos (2 minutos)): `SuspendTimeout` controla a rapidez com que os nós são colocados de volta no sistema e que ficam prontos para uso novamente.
 - Um `SuspendTimeout` menor significa que os nós são redefinidos mais rapidamente e que Slurm pode tentar executar instâncias com mais frequência.
 - Um `SuspendTimeout` mais longo significa que os nós com falha são reinicializados mais lentamente. Enquanto isso, Slurm tenta usar outros nós. Se `SuspendTimeout` demorar mais do que alguns minutos, Slurm tenta percorrer todos os nós do sistema. Um `SuspendTimeout` mais longo pode ser benéfico para sistemas de grande escala (mais de 1.000 nós) para reduzir

o estresse em Slurm enquanto ele tenta enfileirar novamente os trabalhos que falham com frequência.

- Observe que `SuspendTimeout` isso não se refere ao tempo de AWS ParallelCluster espera para encerrar uma instância de apoio para um nó. As instâncias de backup para nós de `POWER_DOWN` são encerradas imediatamente. O processo de encerramento geralmente é concluído em alguns minutos. No entanto, durante esse período, o nó permanece no estado `POWER_DOWN` e não fica disponível para uso do programador.

Logs para a arquitetura

A lista a seguir contém os logs chave. O nome do stream de log usado com o Amazon CloudWatch Logs tem o formato `{hostname}.{instance_id}.{logIdentifier}`, em que `LogIdentifier` segue os nomes dos registros.

- `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`)

Problemas comuns e como depurar:

Nós que falharam ao iniciar, ativar ou ingressar no cluster

- Nós dinâmicos:
 - Verifique o log `ResumeProgram` para ver se `ResumeProgram` foi chamado com o nó. Caso contrário, verifique o log `slurmctld` para determinar se Slurm tentou chamar `ResumeProgram` com o nó. Observe que permissões incorretas em `ResumeProgram` podem fazer com que ele falhe silenciosamente.
 - Se `ResumeProgram` for chamado, verifique se uma instância foi executada para o nó. Se a instância não foi iniciada, deve haver uma mensagem de erro clara sobre o motivo da falha na inicialização da instância.
 - Se uma instância foi iniciada, pode ter havido algum problema durante o processo de bootstrap. Encontre o endereço IP privado e o ID da instância correspondentes no `ResumeProgram`

registro e veja os registros de bootstrap correspondentes para a instância específica em CloudWatch Logs.

- Nós estáticos:
 - Verifique o log `clustermgtd` para ver se foram iniciadas instâncias para o nó. Se as instâncias não iniciarem, deve haver mensagens de erro claras sobre o motivo da falha na inicialização da instância.
 - Se uma instância foi iniciada, houve algum problema com o processo de bootstrap. Encontre o IP privado e o ID da instância correspondentes no `clustermgtd` registro e veja os registros de bootstrap correspondentes para a instância específica em CloudWatch Logs.

Nódulos substituídos ou encerrados inesperadamente e falhas nos nós

- Nós substituídos/encerrados inesperadamente:
 - Na maioria dos casos, `clustermgtd` lida com todas as ações de manutenção do nó. Para verificar se um nó foi `clustermgtd` substituído ou encerrado, verifique o log `clustermgtd`.
 - Se `clustermgtd` tiver substituído ou encerrado o nó, deverá haver uma mensagem indicando o motivo da ação. Se o motivo estiver relacionado com o programador (por exemplo, o nó for DOWN), verifique o log `slurmctld` para obter mais detalhes. Se o motivo estiver relacionado ao EC2, use ferramentas como a Amazon CloudWatch ou o console do AWS EC2, a CLI ou os SDKs para verificar o status ou os registros dessa instância. Por exemplo, você pode verificar se a instância tinha eventos programados ou falhou nas verificações do status de integridade do EC2.
 - Se `clustermgtd` não encerrou o nó, verifique se o nó foi encerrado por `computemgtd` ou se a instância foi encerrada pelo EC2 para recuperar uma instância Spot.
- Falhas do nó:
 - Na maioria dos casos, os trabalhos são automaticamente enfileirados se um nó falhar. Examine o log `slurmctld` para ver por que um trabalho ou um nó falhou e avalie a situação a partir daí.

Falha ao substituir ou encerrar instâncias, falha ao desligar os nós

- Em geral, `clustermgtd` lida com todas as ações esperadas de encerramento da instância. Examine o log `clustermgtd` para ver por que ele não conseguiu substituir ou encerrar um nó.
- Se os nós dinâmicos falharem por [ScaledownIdleTime](#), consulte o log `SuspendProgram` para ver se os processos `slurmctld` fizeram chamadas com o nó específico como argumento. Na verdade, `SuspendProgram` não executa nenhuma ação específica. Em vez disso, ele só cria

logs quando é chamado. Todos os encerramentos de instâncias e redefinições de NodeAddr são concluídos por `clustermgtd`. Slurm faz a transição dos nós para IDLE após `SuspendTimeout`.

Outros problemas:

- AWS ParallelCluster não toma decisões de alocação de tarefas ou escalabilidade. Ele só tenta iniciar, encerrar e manter os recursos de acordo com as instruções do Slurm.

Para problemas relacionados à alocação de trabalhos, alocação de nós e decisão de escalabilidade, consulte o log `slurmctld` em busca de erros.

Modo protegido por cluster do Slurm

Quando um cluster é executado com o modo protegido ativado, o AWS ParallelCluster monitora e rastreia as falhas de bootstrap do nó de computação à medida que os nós de computação são iniciados. Ele faz isso para detectar se essas falhas estão ocorrendo continuamente.

Se o seguinte for detectado em uma fila (partição), o cluster entrará no status protegido:

1. Falhas consecutivas de bootstrap do nó de computação ocorrem continuamente sem nenhuma inicialização bem-sucedida do nó de computação.
2. A contagem de falhas atinge um limite predefinido.

Depois que o cluster entra no status protegido, o AWS ParallelCluster desativa as filas com falhas no limite predefinido ou acima dele.

O modo protegido por cluster do Slurm foi adicionado no AWS ParallelCluster versão 3.0.0.

Você pode usar o modo protegido para reduzir o tempo e os recursos gastos no ciclo de falhas de bootstrap do nó de computação.

Parâmetro do modo protegido

protected_failure_count

`protected_failure_count` especifica o número de falhas consecutivas em uma fila (partição) que ativam o status de cluster protegido.

O padrão `protected_failure_count` é 10 e o modo protegido está ativado.

Se `protected_failure_count` for maior que zero, o modo protegido será ativado.

Se `protected_failure_count` for menor ou igual a zero, o modo protegido será desativado.

Você pode alterar o valor de `protected_failure_count` adicionando o parâmetro no arquivo de configuração `clustermgtd` localizado em `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` no HeadNode.

Você pode atualizar esse parâmetro a qualquer momento e não precisa interromper a frota de computação para fazer isso. Se uma inicialização for bem-sucedida em uma fila antes que a contagem de falhas chegue a `protected_failure_count`, a contagem de falhas será redefinida para zero.

Verificação de status do cluster no status protegido

Quando um cluster está no status protegido, você pode verificar o status da frota de computação e os estados dos nós.

Status da frota de computação

O status da frota de computação é `PROTECTED` em um cluster executado em status protegido.

```
$ pcluster describe-compute-fleet --cluster-name <cluster-name> --region <region-id>
{
  "status": "PROTECTED",
  "lastStatusUpdateTime": "2022-04-22T00:31:24.000Z"
}
```

Status do nó

Para saber quais filas (partições) têm falhas de bootstrap que ativaram o status protegido, faça login no cluster e execute o comando `sinfo`. Partições com falhas de bootstrap iguais ou superiores a `protected_failure_count` ficam no estado `INACTIVE`. Partições sem falhas de bootstrap iguais ou superiores a `protected_failure_count` ficam no estado `UP` e funcionam conforme esperado.

O status `PROTECTED` não afeta os trabalhos em execução. Se os trabalhos estiverem sendo executados em uma partição com falhas de bootstrap iguais ou superiores a `protected_failure_count`, a partição será definida como `INACTIVE` após a conclusão dos trabalhos em execução.

Considere os estados dos nós mostrados no exemplo a seguir.


```
$ 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]
```

A partição queue1 é INACTIVE porque foram detectadas 10 falhas consecutivas de bootstrap do nó de computação.

As instâncias por trás dos nós queue1-dy-c5xlarge-[1-10] foram iniciadas, mas falharam em se juntar ao cluster devido a um status não íntegro.

O cluster está em status protegido.

A partição queue2 não é afetada pelas falhas de bootstrap em queue1. Está no estado UP e ainda pode executar trabalhos.

Como desativar o status protegido

Depois que o erro de bootstrap for resolvido, você poderá executar o comando a seguir para tirar o cluster do status protegido.

```
$ pcluster update-compute-fleet --cluster-name <cluster-name> \
  --region <region-id> \
  --status START_REQUESTED
```

Falhas de bootstrap que ativam o status protegido

Os erros de bootstrap que ativam o status protegido são subdivididos nos três tipos a seguir. Para identificar o tipo e o problema, você pode verificar se o AWS ParallelCluster gerou logs. Se os logs foram gerados, você pode verificá-los para obter detalhes do erro. Para obter mais informações, consulte [Recuperando e preservando logs](#).

1. Erro de bootstrap que faz com que uma instância seja encerrada automaticamente.

Uma instância falha no início do processo de bootstrap, como uma instância que se encerra automaticamente devido a erros no script [SlurmQueues](#) \ [CustomActions](#) \ [OnNodeStart](#) | [OnNodeConfigured](#).

Para nós dinâmicos, procure erros semelhantes aos seguintes:

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

Para nós estáticos, procure no log `clustermgtd (/var/log/parallelcluster/clustermgtd)` por erros semelhantes aos seguintes:

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

2. Os nós `resume_timeout` ou `node_replacement_timeout` expiraram.

Uma instância não pode se juntar ao cluster em `resume_timeout` (para nós dinâmicos) ou `node_replacement_timeout` (para nós estáticos). Ele não encerra automaticamente antes do tempo limite. Por exemplo, a rede não está configurada corretamente para o cluster e o nó é definido para o estado DOWN pelo Slurm após o tempo limite expirar.

Para nós dinâmicos, procure erros semelhantes aos seguintes:

```
Node bootstrap error: Resume timeout expires for node
```

Para nós estáticos, procure no log `clustermgtd (/var/log/parallelcluster/clustermgtd)` por erros semelhantes aos seguintes:

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

3. Os nós falham na verificação de integridade.

Uma instância atrás do nó falha em uma verificação de integridade do EC2 ou na verificação de integridade de um evento agendado, e os nós são tratados como nós de falha de bootstrap. Nesse caso, a instância é encerrada por um motivo fora do controle do AWS ParallelCluster.

Procure no log `clustermgtd (/var/log/parallelcluster/clustermgtd)` por erros semelhantes aos seguintes:

```
Node bootstrap error: Node %s failed during bootstrap when performing health check.
```

4. Os nós de computação falham no registro do Slurm.

O registro do daemon `slurmd` com o daemon de controle do Slurm (`slurmctld`) falha e faz com que o estado do nó de computação mude para o estado `INVALID_REG`. Os nós de computação

do Slurm configurados incorretamente podem causar esse erro, como nós computados configurados com erros de especificação do nó de computação [CustomSlurmSettings](#).

Procure no arquivo de log `slurmctld (/var/log/slurmctld.log)` no nó principal ou no arquivo de log `slurmd (/var/log/slurmd.log)` do nó de computação com falha em busca de erros semelhantes aos seguintes:

```
Setting node %s to INVALID with reason: ...
```

Como depurar o modo protegido

Se seu cluster estiver em status protegido e se AWS ParallelCluster gerou logs `clustermgtd` a partir de `HeadNode` e dos logs de `cloud-init-output` dos nós de computação problemáticos, você poderá verificar os logs para ver os detalhes do erro. Para obter mais informações sobre recuperação de logs, consulte [Recuperando e preservando logs](#).

Log `clustermgtd (/var/log/parallelcluster/clustermgtd)` no nó principal

As mensagens de log mostram quais partições têm falhas de bootstrap e a contagem de falhas de bootstrap correspondente.

```
[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.
```

No log `clustermgtd`, pesquise por `Found the following bootstrap failure nodes` para descobrir qual nó falhou no 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)']
```

No log `clustermgtd`, pesquise por `Node bootstrap error` para descobrir o motivo da falha.

```
[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
```

log `cloud-init-output (/var/log/cloud-init-output.log)` nos nós de computação

Depois de obter o endereço IP privado do nó de falha de bootstrap no log `clustermgtd`, você pode encontrar o log do nó de computação correspondente fazendo login no nó de computação ou seguindo as orientações em [Recuperando e preservando logs](#) para recuperar os logs. Na maioria dos casos, o log `/var/log/cloud-init-output` do nó problemático mostra a etapa que causou a falha de bootstrap do nó de computação.

Failover rápido de capacidade insuficiente do cluster Slurm

A partir do AWS ParallelCluster versão 3.2.0, os clusters são executados com o modo de failover rápido de capacidade insuficiente ativado por padrão. Isso minimiza o tempo gasto tentando colocar novamente um trabalho na fila quando erros de capacidade insuficiente do EC2 são detectados. Isso é particularmente eficaz quando você configura seu cluster com vários tipos de instâncias.

O EC2 detectou falhas de capacidade insuficientes:

- `InsufficientInstanceCapacity`
- `InsufficientHostCapacity`
- `InsufficientReservedInstanceCapacity`
- `MaxSpotInstanceCountExceeded`
- `SpotMaxPriceTooLow`: ativado se o seu preço de solicitação spot é inferior ao preço mínimo de atendimento de solicitação spot exigido.
- `Unsupported`: ativado com o uso de um tipo de instância que não é compatível com uma Região da AWS específica.

No modo rápido de failure-over de capacidade insuficiente, se um erro de capacidade insuficiente for detectado quando um trabalho for atribuído a [SlurmQueues / compute resource](#), o AWS ParallelCluster faz o seguinte:

1. Ele define o recurso de computação para um estado desativado (DOWN) por um período de tempo predefinido.
2. Ele usa `POWER_DOWN_FORCE` para cancelar as tarefas do nó com falha do recurso de computação e para suspender o nó com falha. Ele define o nó com falha para o estado `IDLE` e `POWER_DOWN (!)` e, em seguida, para `POWERING_DOWN (%)`.
3. Ele enfileira o trabalho em outro recurso de computação.

Os nós estáticos e ativados do recurso de computação desativado não são afetados. Os trabalhos podem ser concluídos nesses nós.

Esse ciclo se repete até que o trabalho seja atribuído com êxito a um ou mais nós de recursos de computação. Para mais informações sobre esses estados, consulte [Guia do Slurm para o modo de várias filas](#).

Se nenhum recurso de computação for encontrado para executar o trabalho, o trabalho será definido no estado PENDING até que o período de tempo predefinido termine. Nesse caso, você pode modificar o período de tempo predefinido conforme descrito na seção a seguir.

Parâmetro de tempo limite de capacidade insuficiente

insufficient_capacity_timeout

`insufficient_capacity_timeout` especifica o período de tempo (em segundos) em que o recurso de computação é mantido no estado desativado (down) quando um erro de capacidade insuficiente é detectado.

Por padrão, `insufficient_capacity_timeout` é habilitado.

O `insufficient_capacity_timeout` padrão é 600 segundos (10 minutos).

Se o valor `insufficient_capacity_timeout` for menor ou igual a zero, o modo de failover rápido de capacidade insuficiente será desativado.

Você pode alterar o valor de `insufficient_capacity_timeout` adicionando o parâmetro no arquivo de configuração `clustermgtd` localizado em `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` no HeadNode.

O parâmetro pode ser atualizado a qualquer momento, sem interromper a frota de computação.

Por exemplo:

- `insufficient_capacity_timeout=600:`

Se um erro de capacidade insuficiente for detectado, o recurso de computação será definido como desativado (DOWN). Após 10 minutos, seu nó com falha é definido para o estado `idle~ (POWER_SAVING)`.

- `insufficient_capacity_timeout=60:`

Se um erro de capacidade insuficiente for detectado, o recurso de computação será definido como desativado (DOWN). Após 1 minuto, seu nó com falha é definido para o estado `idle~`.

- `insufficient_capacity_timeout=0`:

O modo de failover rápido de capacidade insuficiente é desativado. O recurso de computação não é desativado.

Note

Pode haver um atraso de até um minuto entre o momento em que os nós falham com erros de capacidade insuficientes e o momento em que o daemon de gerenciamento de cluster detecta as falhas do nó. Isso ocorre porque o daemon de gerenciamento de cluster verifica falhas de capacidade insuficiente do nó e define os recursos de computação para o estado `down` em intervalos de um minuto.

Status do modo de failover rápido de capacidade insuficiente

Quando um cluster está no modo de failover rápido e de capacidade insuficiente, você pode verificar seu status e os estados dos nós.

Estados do nó

Quando uma tarefa é enviada a um nó dinâmico de recursos de computação e um erro de capacidade insuficiente é detectado, o nó é colocado nesse estado `down#` com razão.

```
(Code:InsufficientInstanceCapacity)Failure when resuming nodes.
```

Depois os nós desligados (nós no estado `idle~`) são configurados para `down~` com razão.

```
(Code:InsufficientInstanceCapacity)Temporarily disabling node due to insufficient capacity.
```

O trabalho é enfileirado para outros recursos de computação na fila.

Os nós estáticos do recurso de computação e os nós que são UP não são afetados pelo modo de failover rápido de capacidade insuficiente.

Considere os estados dos nós mostrados no exemplo a seguir.

```
$ 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]
```

Enviamos um trabalho para a queue1, que requer um nó.

```
$ 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]
```

O nó queue1-dy-c-1-1 é iniciado para executar o trabalho. No entanto, a instância falhou ao ser iniciada devido a um erro de capacidade insuficiente. O nó queue1-dy-c-1-1 está definido como down. O nó dinâmico desligado dentro do recurso de computação (queue2-dy-c-1) está definido como down.

Você pode verificar a razão do nó com `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]
```

O trabalho é enfileirado em outro tipo de instância dentro dos recursos de computação da fila.

Depois de decorrido o `insufficient_capacity_timeout`, os nós no recurso de computação são redefinidos para o estado `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]
```

Depois de decorrido o `insufficient_capacity_timeout` e os nós no recurso de computação são redefinidos para o estado `idle~`, o programador Slurm atribui prioridade menor aos nós. O programador continua selecionando nós de outros recursos de computação da fila com pesos maiores, a menos que ocorra uma das seguintes situações:

- Os requisitos de envio de um trabalho correspondem ao recurso de computação recuperado.
- Nenhum outro recurso de computação está disponível porque eles estão lotados.
- `slurmctld` é reiniciado.
- A frota de computação do AWS ParallelCluster é interrompida e começa a desligar e ligar todos os nós.

Logs relacionados

Os logs relacionados a erros de capacidade insuficiente e ao modo rápido de failover de capacidade insuficiente podem ser encontrados no log `resume` do Slurm e no log `clustermgtd` no nó principal.

Slurm `resume` (`/var/log/parallelcluster/slurm_resume.log`)

Mensagens de erro quando um nó falha na inicialização devido à capacidade insuficiente.

```
[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`)

O recurso de computação `c-1` em `queue1` está desativado devido à capacidade insuficiente.


```
[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
```

Depois que o tempo limite de capacidade insuficiente expirar, o recurso de computação é redefinido e os nós dentro dos recursos de computação são definidos como `idle~`.

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the following compute resources because insufficient capacity timeout expired: {'queue1': ['c-1']}
```

Programação baseada em memória do Slurm

A partir da versão 3.2.0, AWS ParallelCluster oferece suporte ao agendamento Slurm baseado em memória com o parâmetro de configuração [SlurmSettingsEnableMemoryBasedScheduling/cluster](#).

Note

[A partir da AWS ParallelCluster versão 3.7.0, EnableMemoryBasedScheduling pode ser ativado se você configurar vários tipos de instância em Instâncias.](#)

Para AWS ParallelCluster as versões 3.2.0 a 3.6. x, não EnableMemoryBasedScheduling pode ser ativado se você configurar vários tipos de instância em [Instâncias](#).

Warning

Quando você especifica vários tipos de instâncias em um recurso de computação de fila Slurm com EnableMemoryBasedScheduling ativado, o valor RealMemory é a quantidade mínima de memória disponibilizada para todos os tipos de instância. Isso pode resultar em quantidades significativas de memória não utilizada se você especificar tipos de instância com capacidades de memória muito diferentes.

Com `EnableMemoryBasedScheduling: true`, o programador do Slurm rastreia a quantidade de memória que cada trabalho exige em cada nó. Em seguida, o programador do Slurm usa essas informações para agendar vários trabalhos no mesmo nó de computação. A quantidade total de memória que os trabalhos exigem em um nó não pode ser maior do que a memória disponível do nó. O programador impede que um trabalho use mais memória do que a solicitada quando o trabalho foi enviado.

Com `EnableMemoryBasedScheduling: false`, os trabalhos podem competir pela memória em um nó compartilhado e causar falhas no trabalho e eventos `out-of-memory`.

Warning

O Slurm usa uma notação de potência de 2 para seus rótulos, como MB ou GB. Leia esses rótulos como MiB e GiB, respectivamente.

Configuração do Slurm e programação baseada em memória

Com `EnableMemoryBasedScheduling: true`, o Slurm define os seguintes parâmetros de configuração do Slurm:

- [SelectTypeParameters=CR_CPU_Memory](#) no `slurm.conf`. Essa opção configura a memória do nó para ser um recurso consumível no Slurm.
- [ConstrainRAMSpace=yes](#) no `cgroup.conf` do Slurm. Com essa opção, o acesso de um trabalho à memória é limitado à quantidade de memória que o trabalho solicitou quando enviado.

Note

Vários outros parâmetros de configuração do Slurm podem afetar o comportamento do programador do Slurm e do gerenciador de recursos quando essas duas opções são definidas. Para obter mais informações, consulte a [Documentação do Slurm](#).

Programador do Slurm e programação baseada em memória

EnableMemoryBasedScheduling: false (padrão)

Por padrão, `EnableMemoryBasedScheduling` é definido como falso. Quando falso, o Slurm não inclui memória como recurso em seu algoritmo de agendamento e não rastreia a memória que os

trabalhos usam. Os usuários podem especificar a opção `--mem MEM_PER_NODE` para definir a quantidade mínima de memória por nó que um trabalho exige. Isso força o programador a escolher nós com um valor `RealMemory` de pelo menos `MEM_PER_NODE` ao agendar o trabalho.

Por exemplo, suponha que um usuário envie dois trabalhos com `--mem=5GB`. Se os recursos solicitados, como CPUs ou GPUs, estiverem disponíveis, os trabalhos poderão ser executados ao mesmo tempo em um nó com 8 GiB de memória. As duas tarefas não estão programadas em nós de computação com menos de 5 GiB de `RealMemory`.

Warning

Quando a programação baseada em memória está desativada, o Slurm não rastreia a quantidade de memória que os trabalhos usam. Os trabalhos executados no mesmo nó podem competir por recursos de memória e fazer com que o outro trabalho falhe.

Quando a programação baseada em memória está desativada, recomendamos que os usuários não especifiquem as opções `--mem-per-cpu` ou `--mem-per-gpu`. Essas opções podem causar um comportamento diferente do descrito na documentação do [Slurm](#).

EnableMemoryBasedScheduling: true

Quando `EnableMemoryBasedScheduling` é definido como verdadeiro, o Slurm rastreia o uso da memória de cada trabalho e impede que os trabalhos usem mais memória do que a solicitada com as opções de envio `--mem`.

Usando o exemplo anterior, um usuário envia dois trabalhos com `--mem=5GB`. Os trabalhos não podem ser executados ao mesmo tempo em um nó com 8 GiB de memória. Isso ocorre porque a quantidade total de memória necessária é maior do que a memória disponível no nó.

Com a programação baseada em memória ativada, `--mem-per-cpu` e `--mem-per-gpu` se comportam de forma consistente com o que está descrito na documentação do Slurm. Por exemplo, um trabalho é enviado com `--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB`. Nesse caso, Slurm atribui à tarefa um total de 4 GiB para cada nó.

Warning

Quando a programação baseada em memória está ativada, recomendamos que os usuários incluam uma especificação `--mem` ao enviar um trabalho. Com a Slurm configuração padrão

incluída em AWS ParallelCluster, se nenhuma opção de memória estiver incluída (`--mem,` ou `--mem-per-gpu`)-`--mem-per-cpu`, Slurm atribui toda a memória dos nós alocados à tarefa, mesmo que ela solicite somente uma parte dos outros recursos, como CPUs ou GPUs. Isso evita efetivamente o compartilhamento de nós até que o trabalho seja concluído, pois não há memória disponível para outros trabalhos. Isso acontece porque o Slurm define a memória por nó da tarefa para [DefMemPerNode](#) quando nenhuma especificação de memória é fornecida no momento do envio da tarefa. O valor padrão para esse parâmetro é 0 e especifica o acesso ilimitado à memória de um nó.

Se vários tipos de recursos de computação com quantidades diferentes de memória estiverem disponíveis na mesma fila, um trabalho enviado sem opções de memória poderá receber quantidades diferentes de memória em nós diferentes. Isso depende de quais nós o programador disponibiliza para o trabalho. Os usuários podem definir um valor personalizado para opções, como `DefMemPerNode` ou [DefMemPerCPU](#), no nível do cluster ou da partição nos arquivos de configuração do Slurm para evitar esse comportamento.

Slurm `RealMemory` e AWS ParallelCluster `SchedulableMemory`

Com a Slurm configuração fornecida AWS ParallelCluster, Slurm interpreta como [RealMemory](#) quantidade de memória por nó disponível para trabalhos. A partir da versão 3.2.0, por padrão, AWS ParallelCluster define `RealMemory` 95% da memória listada nos tipos de [instância do Amazon EC2](#) e retornada pela API do Amazon EC2. [DescribeInstanceTypes](#)

Quando a programação baseada em memória é desativada, o Slurm programador usa os nós do `RealMemory` para filtrar os nós quando os usuários enviam um trabalho com `--mem` especificado.

Quando a programação baseada em memória está ativada, o programador do Slurm interpreta como `RealMemory` a quantidade máxima de memória disponível para trabalhos em execução no nó de computação.

A configuração padrão pode não ser ideal para todos os tipos de instância:

- Essa configuração pode ser maior do que a quantidade de memória que os nós podem realmente acessar. Isso pode acontecer quando os nós de computação são tipos de instâncias pequenas.
- Essa configuração pode ser maior do que a quantidade de memória que os nós podem realmente acessar. Isso pode acontecer quando os nós de computação são tipos de instância grandes e podem levar a uma quantidade significativa de memória não utilizada.

Você pode usar [SlurmQueues/ComputeResources/SchedulableMemory](#) para ajustar o valor de `RealMemory` configure by AWS ParallelCluster para nós de computação. Para substituir o padrão, defina um valor personalizado para `SchedulableMemory` especificamente para sua configuração de cluster.

Para verificar a memória real disponível de um nó de computação, execute o comando `/opt/slurm/sbin/slurmd -C` no nó. Esse comando retorna a configuração de hardware do nó, incluindo o valor [RealMemory](#). Para ter mais informações, consulte [slurmd -C](#).

Certifique-se de que os processos do sistema operacional do nó de computação tenham memória suficiente. Para fazer isso, limite a memória disponível para trabalhos definindo o valor de `SchedulableMemory` como menor do que o valor de `RealMemory` retornado pelo comando `slurmd -C`.

Alocação a vários tipos de instância com o Slurm

A partir da versão 3.3.0 do AWS ParallelCluster, você pode configurar seu cluster para alocar a partir de um conjunto de tipos de instância definidos em um recurso de computação. A alocação pode ser baseada em estratégias capacidade ideal ou de baixo custo de frota do EC2.

Esse conjunto de tipos de instância definidos deve ter o mesmo número de vCPUs em todos ou, se o multithreading estiver desativado, o mesmo número de núcleos. Além disso, esse conjunto de tipos de instância deve ter o mesmo número de aceleradores dos mesmos fabricantes. Se `Efa / Enabled` estiver definido como `true`, as instâncias devem ter suporte ao EFA. Para obter mais informações e requisitos, consulte [Scheduling / SlurmQueues / AllocationStrategy](#) e [ComputeResources / Instances](#).

Você pode definir [AllocationStrategy](#) para `lowest-price` ou `capacity-optimized` dependendo da configuração [CapacityType](#).

Em [Instances](#), você pode configurar um conjunto de tipos de instância.

Note

A partir da versão 3.7.0 do AWS ParallelCluster, `EnableMemoryBasedScheduling` pode ser ativado se você configurar vários tipos de instância em [Instâncias](#).

Para as versões 3.2.0 a 3.6.x do AWS ParallelCluster, `EnableMemoryBasedScheduling` não pode ser ativado se você configurar vários tipos de instância em [Instâncias](#).

Os exemplos a seguir mostram como você pode consultar tipos de instância para vCPUs, suporte ao EFA e arquitetura.

Consulta InstanceTypes com 96 vCPUs e arquitetura 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,  
&InstanceType})" \  
  --output table
```

Consulta InstanceTypes com 64 núcleos, suporte ao EFA e arquitetura 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,  
&InstanceType})" \  
  --output table
```

O próximo exemplo de trecho de configuração de cluster mostra como você pode usar essas propriedades InstanceType e AllocationStrategy.

```
...  
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
...
```

Dimensionamento de clusters para nós dinâmicos

ParallelCluster suporta Slurm métodos para escalar dinamicamente clusters usando o plug-in Slurm de economia de energia. Para obter mais informações, consulte o [Cloud Scheduling Guide](#) e o [Power Saving Guide do Slurm](#) na documentação do Slurm.

A partir da ParallelCluster versão 3.8.0, ParallelCluster usa o currículo no nível do trabalho ou o escalonamento no nível do trabalho como a estratégia padrão de alocação dinâmica de nós para ParallelCluster escalar o cluster: expande o cluster com base nos requisitos de cada trabalho, no número de nós alocados para o trabalho e nos nós que precisam ser retomados. ParallelCluster obtém essas informações da variável de ambiente SLURM_RESUME_FILE.

O escalonamento para nós dinâmicos é um processo de duas etapas, que envolve o lançamento das instâncias do EC2 e a atribuição das instâncias do EC2 lançadas aos nós do Slurm. Cada uma dessas duas etapas pode ser executada usando uma lógica all-or-nothing de melhor esforço.

Para o lançamento das instâncias do EC2:

- all-or-nothing chama a API de lançamento do EC2 com uma meta mínima igual à capacidade total de destino
- A best effort chama a API de lançamento do EC2 com a meta mínima igual a 1 e a capacidade alvo total igual à capacidade solicitada

Para atribuição das instâncias do EC2 aos nós do Slurm:

- all-or-nothing atribui instâncias do EC2 aos nós do Slurm somente se for possível atribuir uma instância do EC2 a cada nó solicitado
- best-effort atribui instâncias do EC2 aos nós do Slurm, mesmo que todos os nós solicitados não estejam cobertos pela capacidade da instância do EC2

As combinações possíveis das estratégias acima se traduzem nas estratégias de ParallelCluster lançamento.

Example

<caption>The available ParallelCluster estratégias de lançamento that can be set into the [ScalingStrategy](#) cluster configuration to be used with escalabilidade em nível de trabalho are:</caption>

all-or-nothingdimensionamento:

Essa estratégia envolve AWS ParallelCluster iniciar uma chamada de API de instância de lançamento do Amazon EC2 para cada trabalho, o que exige que todas as instâncias necessárias para que os nós de computação solicitados sejam executados com sucesso. Isso garante que o cluster seja escalado somente quando a capacidade necessária por trabalho estiver disponível, evitando instâncias ociosas deixadas no final do processo de escalabilidade.

A estratégia usa uma all-or-nothinglógica para o lançamento das instâncias do EC2 para cada trabalho e uma all-or-nothinglógica para a atribuição das instâncias do EC2 aos nós do Slurm.

Os grupos de estratégias iniciam solicitações em lotes, uma para cada recurso computacional solicitado e até 500 nós cada. Para solicitações que abrangem vários recursos computacionais ou excedem 500 nós, processa ParallelCluster sequencialmente vários lotes.

A falha do lote de um único recurso resulta no encerramento de toda a capacidade não utilizada associada, garantindo que nenhuma instância ociosa seja deixada no final do processo de escalabilidade.

Limitações

- O tempo necessário para escalar é diretamente proporcional ao número de trabalhos enviados por execução do programa de currículo do Slurm.
- A operação de escalabilidade é limitada pelo limite da conta de RunInstances recursos, definido em 1.000 instâncias por padrão. Essa limitação está de acordo com as políticas de limitação de API AWS do EC2. Para obter mais detalhes, consulte a documentação de limitação da API do [AWS EC2](#)
- Quando você envia um trabalho em um recurso computacional com um único tipo de instância, em uma fila que abrange várias zonas de disponibilidade, a chamada da API de inicialização do all-or-nothingEC2 só é bem-sucedida se toda a capacidade puder ser fornecida em uma única zona de disponibilidade.
- Quando você envia um trabalho em um recurso computacional com vários tipos de instância, em uma fila com uma única zona de disponibilidade, a chamada da API de inicialização do all-or-

nothingEC2 só é bem-sucedida se toda a capacidade puder ser fornecida por um único tipo de instância.

- Quando você envia um trabalho em um recurso computacional com vários tipos de instância, em uma fila que abrange várias zonas de disponibilidade, a chamada da API de lançamento do all-or-nothingEC2 não é suportada e, em vez disso, ParallelCluster executa o escalonamento da melhor maneira possível.

greedy-all-or-nothingdimensionamento:

Essa variante da all-or-nothing estratégia ainda garante que o cluster seja escalado somente quando a capacidade necessária por trabalho estiver disponível, evitando instâncias ociosas no final do processo de escalabilidade, mas envolve ParallelCluster iniciar uma chamada de API de instância de lançamento do Amazon EC2 que visa uma capacidade mínima de destino de 1, tentando maximizar o número de nós lançados até a capacidade solicitada. A estratégia usa uma lógica de melhor esforço para o lançamento das instâncias do EC2 para todos os trabalhos, além da all-or-nothinglógica para a atribuição das instâncias do EC2 aos nós do Slurm para cada trabalho.

Os grupos de estratégias iniciam solicitações em lotes, uma para cada recurso computacional solicitado e até 500 nós cada. Para solicitações que abrangem vários recursos computacionais ou excedem 500 nós, o Parallelcluster processa sequencialmente vários lotes.

Ele garante que nenhuma instância ociosa seja deixada no final do processo de escalabilidade, maximizando a taxa de transferência ao custo da superescalabilidade temporária durante o processo de escalabilidade.

Limitações

- A superescalabilidade temporária é possível, gerando custos adicionais para instâncias que fazem a transição para um estado de execução antes da conclusão do escalonamento.
- O mesmo limite de instância da all-or-nothing estratégia se aplica, sujeito ao limite AWS da conta de RunInstances recursos da.

escalabilidade com o melhor esforço:

Essa estratégia chama a chamada da API de instância de lançamento do EC2 visando uma capacidade mínima de 1 e visando atingir a capacidade total solicitada ao custo de deixar instâncias ociosas após a execução do processo de escalabilidade, caso nem toda a capacidade solicitada esteja disponível. A estratégia usa uma lógica de melhor esforço para o lançamento das instâncias

do EC2 para todos os trabalhos, além da lógica de melhor esforço para a atribuição das instâncias do Amazon EC2 aos nós do Slurm para cada trabalho.

Os grupos de estratégias iniciam solicitações em lotes, uma para cada recurso computacional solicitado e até 500 nós cada. Para solicitações que abrangem vários recursos computacionais ou excedem 500 nós, processa ParallelCluster sequencialmente vários lotes.

Essa estratégia permite escalar muito além do limite padrão de 1.000 instâncias em várias execuções de processos de escalabilidade, ao custo de ter instâncias ociosas nos diferentes processos de escalabilidade.

Limitações

- Possíveis instâncias em execução ociosas no final do processo de escalabilidade, caso não seja possível alocar todos os nós solicitados pelos trabalhos.

Veja a seguir um exemplo que mostra como o dimensionamento dos nós dinâmicos se comporta usando as diferentes estratégias de ParallelCluster lançamento. Suponha que você tenha enviado dois trabalhos solicitando 20 nós cada, totalizando 40 nós do mesmo tipo, mas há apenas 30 instâncias do EC2 disponíveis para cobrir a capacidade solicitada no EC2.

all-or-nothing dimensionamento:

- Para o primeiro trabalho, uma API de instância de execução do all-or-nothing EC2 é chamada, solicitando 20 instâncias. Uma chamada bem-sucedida resulta no lançamento de 20 instâncias
- all-or-nothing a atribuição das 20 instâncias lançadas aos nós do Slurm para o primeiro trabalho foi bem-sucedida
- Outra API de instância de execução do all-or-nothing EC2 é chamada, solicitando 20 instâncias para o segundo trabalho. A chamada não foi bem-sucedida, pois só há capacidade para outras 10 instâncias. Nenhuma instância foi iniciada no momento

greedy-all-or-nothing dimensionamento:

- A API de instância de lançamento do EC2 de melhor esforço é chamada, solicitando 40 instâncias, que é a capacidade total solicitada por todos os trabalhos. Isso resulta no lançamento de 30 instâncias
- Uma all-or-nothing atribuição de 20 das instâncias lançadas aos nós do Slurm para o primeiro trabalho foi bem-sucedida.

- Outra all-or-nothing atribuição das instâncias lançadas restantes aos nós do Slurm para o segundo trabalho foi tentada, mas como há apenas 10 instâncias disponíveis do total de 20 solicitadas pelo trabalho, a atribuição não foi bem-sucedida
- As 10 instâncias lançadas não atribuídas são encerradas

escalabilidade com o melhor esforço:

- A API de instância de lançamento do EC2 de melhor esforço é chamada, solicitando 40 instâncias, que é a capacidade total solicitada por todos os trabalhos. Isso resulta no lançamento de 30 instâncias.
- A atribuição máxima de 20 das instâncias lançadas aos nós do Slurm para o primeiro trabalho foi bem-sucedida.
- Outra atribuição de maior esforço das 10 instâncias lançadas restantes aos nós do Slurm para o segundo trabalho é bem-sucedida, mesmo que a capacidade total solicitada tenha sido 20. Porém, como o trabalho estava solicitando os 20 nós e foi possível atribuir instâncias do EC2 a apenas 10 deles, o trabalho não pode ser iniciado e as instâncias ficam inativas, até que seja encontrada capacidade suficiente para iniciar as 10 instâncias ausentes em uma chamada posterior do processo de escalabilidade, ou o programador agenda o trabalho em outros nós de computação já em execução.

Slurmestratégias dinâmicas de alocação de nós na versão 3.7.x

ParallelCluster usa dois tipos de estratégias dinâmicas de alocação de nós para escalar o cluster:

- Alocação com base nas informações disponíveis do nó solicitado:
 - Retomada de todos os nós ou escalabilidade da lista de nós:

ParallelCluster aumenta o cluster com base somente nos nomes Slurm da lista de nós solicitada quando Slurm ele é `ResumeProgram` executado. Ele aloca recursos computacionais aos nós somente pelo nome do nó. A lista de nomes de nós pode se estender por vários trabalhos.

- Continuação em nível de trabalho ou escalonamento em nível de trabalho:

ParallelCluster aumenta o cluster com base nos requisitos de cada trabalho, no número atual de nós alocados para o trabalho e nos nós que precisam ser retomados. ParallelCluster obtém essas informações da variável de `SLURM_RESUME_FILE` ambiente.

- Alocação com uma estratégia de execução do EC2:

- Escalabilidade com o melhor esforço:

ParallelCluster expande o cluster usando uma chamada de API de instância de execução do EC2 com a capacidade mínima de destino igual a 1, para iniciar algumas, mas não necessariamente todas, as instâncias necessárias para dar suporte aos nós solicitados.

- Uma ll-or-nothing escala:

ParallelCluster expande o cluster usando uma chamada de API de instância de execução do EC2 que só é bem-sucedida se todas as instâncias necessárias para dar suporte aos nós solicitados forem iniciadas. Nesse caso, ele chama a API da instância de execução do EC2 com a capacidade mínima de destino igual à capacidade total solicitada.

Por padrão, ParallelCluster usa escalabilidade de lista de nós com a melhor estratégia de lançamento do EC2 para lançar algumas, mas não necessariamente todas, as instâncias necessárias para dar suporte aos nós solicitados. Ele tenta provisionar o máximo de capacidade possível para atender ao workload enviado.

A partir da ParallelCluster versão 3.7.0, ParallelCluster usa escalabilidade em nível de trabalho com uma estratégia de lançamento do all-or-nothingEC2 para trabalhos enviados no modo exclusivo. Quando você envia um trabalho em modo exclusivo, o trabalho tem acesso exclusivo aos nós alocados. Para obter mais informações, consulte [EXCLUSIVE](#) na documentação do Slurm.

Para enviar um trabalho no modo exclusivo:

- Transmita o sinalizador exclusivo ao enviar um trabalho do Slurm para o cluster. Por exemplo, `sbatch ... --exclusive`.

OU

- Envie um trabalho para uma fila de cluster que tenha sido configurada com [JobExclusiveAllocation](#) definido como `true`.

Ao enviar um trabalho no modo exclusivo:

- ParallelCluster atualmente agrupa solicitações de lançamento em lotes para incluir até 500 nós. Se um trabalho solicitar mais de 500 nós, ParallelCluster faz uma solicitação de all-or-nothinginicialização para cada conjunto de 500 nós e uma solicitação de inicialização adicional para o restante dos nós.

- Se a alocação de nós estiver em um único recurso computacional, ParallelCluster fará uma solicitação de all-or-nothing inicialização para cada conjunto de 500 nós e uma solicitação de inicialização adicional para o restante dos nós. Se uma solicitação de lançamento falhar, ParallelCluster encerrará a capacidade não utilizada criada por todas as solicitações de lançamento.
- Se a alocação de nós abranger vários recursos computacionais, será ParallelCluster necessário fazer uma solicitação de all-or-nothing lançamento para cada recurso computacional. Essas solicitações também são agrupadas. Se uma solicitação de inicialização falhar para um dos recursos computacionais, ParallelCluster a capacidade não utilizada criada por todas as solicitações de inicialização do recurso computacional será encerrada.

escalabilidade em nível de trabalho com limitações conhecidas da estratégia de all-or-nothing lançamento:

- Quando você envia um trabalho em um recurso computacional com um único tipo de instância, em uma fila que abrange várias zonas de disponibilidade, a chamada da API de inicialização do all-or-nothing EC2 só é bem-sucedida se toda a capacidade puder ser fornecida em uma única zona de disponibilidade.
- Quando você envia um trabalho em um recurso computacional com vários tipos de instância, em uma fila com uma única zona de disponibilidade, a chamada da API de inicialização do all-or-nothing EC2 só é bem-sucedida se toda a capacidade puder ser fornecida por um único tipo de instância.
- Quando você envia um trabalho em um recurso computacional com vários tipos de instância, em uma fila que abrange várias zonas de disponibilidade, a chamada da API de lançamento do all-or-nothing EC2 não é suportada e, em vez disso, ParallelCluster executa o escalonamento da melhor maneira possível.

Slurm estratégias dinâmicas de alocação de nós na versão 3.6.x e anteriores

AWS ParallelCluster usa somente um tipo de estratégia dinâmica de alocação de nós para escalar o cluster:

- Alocação com base nas informações disponíveis do nó solicitado:
 - Reinício de todos os nós ou escalabilidade da lista de nós: ParallelCluster expande o cluster com base somente nos nomes Slurm da lista de nós solicitada quando ele é executado. Slurm

`ResumeProgram` Ele aloca recursos computacionais aos nós somente pelo nome do nó. A lista de nomes de nós pode se estender por vários trabalhos.

- Alocação com uma estratégia de execução do EC2:
 - Escalonamento com o melhor esforço: ParallelCluster expande o cluster usando uma chamada de API de instância de execução do EC2 com a capacidade mínima de destino igual a 1, para iniciar algumas, mas não necessariamente todas, as instâncias necessárias para dar suporte aos nós solicitados.

ParallelCluster usa o escalonamento da lista de nós com a melhor estratégia de lançamento do EC2 para lançar algumas, mas não necessariamente todas, as instâncias necessárias para dar suporte aos nós solicitados. Ele tenta provisionar o máximo de capacidade possível para atender ao workload enviado.

Limitações

- Possíveis instâncias em execução ociosas no final do processo de escalabilidade, caso não seja possível alocar todos os nós solicitados pelos trabalhos.

Slurmcontabilidade com AWS ParallelCluster

A partir da versão 3.3.0, AWS ParallelCluster oferece suporte à Slurm contabilidade com o parâmetro de configuração de cluster [SlurmSettings/Database](#).

Com a contabilização de Slurm, você pode integrar um banco de dados contábil externo para fazer o seguinte:

- Gerenciar usuários do cluster ou grupos de usuários e outras entidades. Com esse recurso, você pode usar os recursos mais avançados do Slurm, como imposição de limite de recursos, fairshare e QOS.
- Coleta e salva dados do trabalho, como o usuário que executou o trabalho, a duração do trabalho e os recursos que ele usa. Você pode visualizar os dados salvos com o utilitário `sacct`.

Note

AWS ParallelCluster suporta a Slurm contabilização de [Slurmservidores de banco de dados MySQL compatíveis](#).

Trabalhando com Slurm contabilidade em AWS ParallelCluster

Antes de configurar a contabilização de Slurm, você deve ter um servidor de banco de dados externo existente e um banco de dados que use um protocolo mysql.

Para configurar a Slurm contabilidade com AWS ParallelCluster, você deve definir o seguinte:

- O URI do servidor de banco de dados externo em [Database/Uri](#). O servidor deve existir e estar acessível a partir do nó principal.
- Credenciais para acessar o banco de dados externo que estão definidas em [Database/PasswordSecretArne Database/UserName](#). AWS ParallelCluster usa essas informações para configurar a contabilidade no Slurm nível e o `slurmdbd` serviço no nó principal. `slurmdbd` é o daemon que gerencia a comunicação entre o cluster e o servidor do banco de dados.

Para ver um tutorial, consulte [Criação de um cluster com contabilidade Slurm](#).

Note

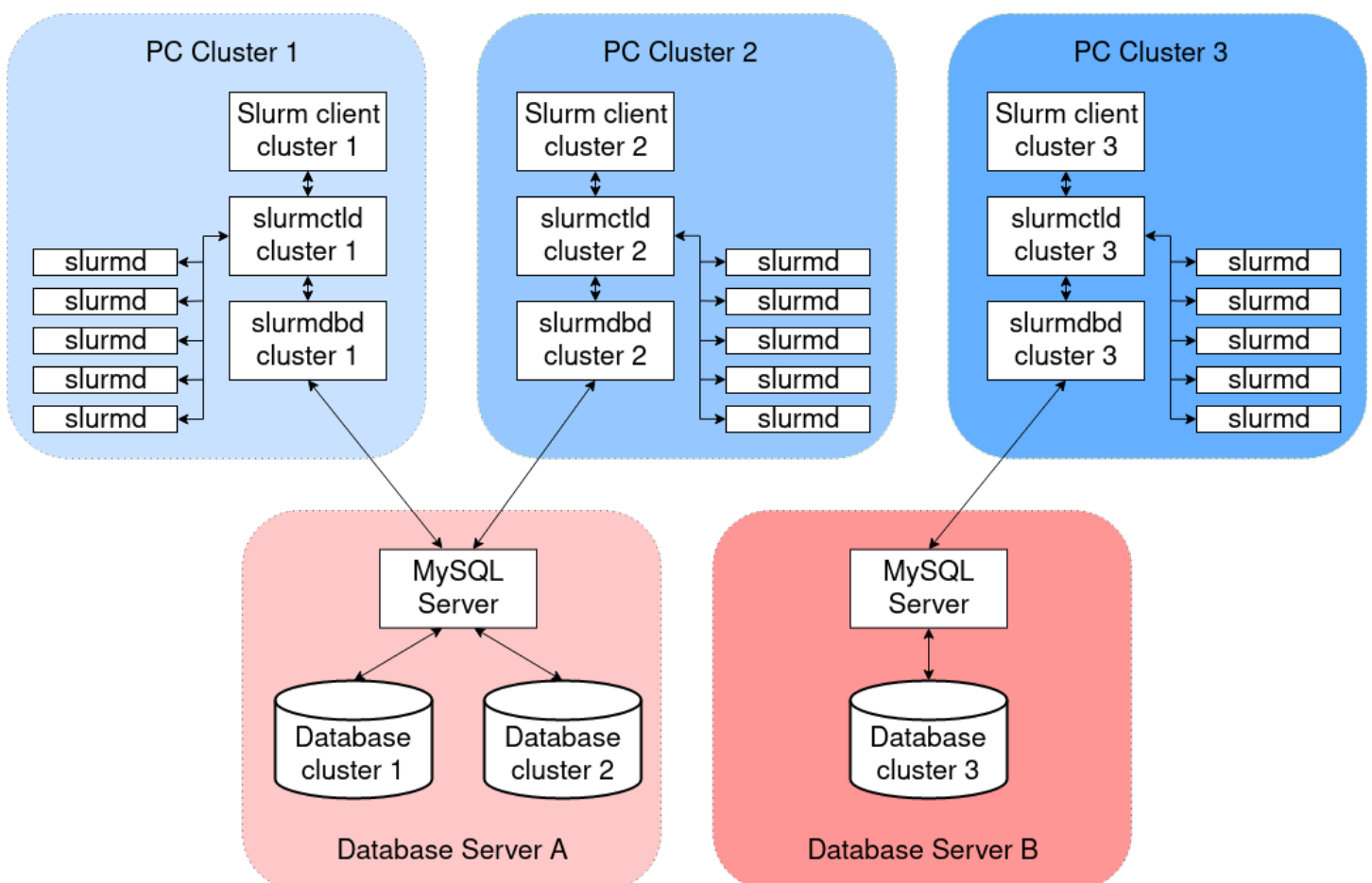
AWS ParallelCluster executa um bootstrap básico do banco de dados Slurm contábil definindo o usuário padrão do cluster como administrador do banco de dados no Slurm banco de dados. AWS ParallelCluster não adiciona nenhum outro usuário ao banco de dados contábil. O cliente é responsável por gerenciar as entidades de contabilização no banco de dados Slurm.

AWS ParallelCluster configura `slurmdbd` para garantir que um cluster tenha seu próprio Slurm banco de dados no servidor de banco de dados. O mesmo servidor de banco de dados pode ser usado em vários clusters, mas cada cluster tem seu próprio banco de dados separado. AWS ParallelCluster usa o nome do cluster para definir o nome do banco de dados no `StorageLoc` parâmetro do arquivo de `slurmdbd` configuração. Considere a situação a seguir. Um banco de dados que está presente no servidor de banco de dados inclui um nome de cluster que não é mapeado para um nome de cluster ativo. Nesse caso, você pode criar um novo cluster com esse nome de cluster para mapear para esse banco de dados. Slurm reutiliza o banco de dados para o novo cluster.

⚠ Warning

- Não recomendamos configurar mais de um cluster para usar o mesmo banco de dados ao mesmo tempo. Isso pode levar a problemas de desempenho ou até mesmo a situações de impasse no banco de dados.
- Se a contabilização de Slurm estiver habilitada no nó principal de um cluster, recomendamos usar um tipo de instância com uma CPU poderosa, mais memória e maior largura de banda da rede. A contabilização de Slurm pode sobrecarregar o nó principal do cluster.

Na arquitetura atual do recurso de AWS ParallelCluster Slurm contabilidade, cada cluster tem sua própria instância do `slurmdbd` daemon, conforme mostrado nos exemplos de configurações do diagrama a seguir.



Se você estiver adicionando funcionalidades personalizadas de vários clusters Slurm ou de federação ao seu ambiente de cluster, todos os clusters devem fazer referência à mesma instância `slurmdbd`. Para essa alternativa, recomendamos que você habilite a AWS ParallelCluster Slurm contabilização em um cluster e configure manualmente os outros clusters para se conectarem aos `slurmdbd` que estão hospedados no primeiro cluster.

Se você estiver usando AWS ParallelCluster versões anteriores à versão 3.3.0, consulte o método alternativo para implementar a Slurm contabilidade descrito nesta postagem do [blog sobre HPC](#).

Considerações sobre contabilização de Slurm

Banco de dados e cluster em diferentes VPCs

Para habilitar a contabilização de Slurm, é necessário um servidor de banco de dados para servir como back-end para as operações de leitura e gravação que o daemon `slurmdbd` executa. Antes que o cluster seja criado ou atualizado para permitir a contabilização de Slurm, o nó principal deve ser capaz de acessar o servidor do banco de dados.

Se você precisar implantar o servidor de banco de dados em uma VPC diferente da usada pelo cluster, considere o seguinte:

- Para permitir a comunicação entre o `slurmdbd` do lado do cluster e do servidor do banco de dados, você deve configurar a conectividade entre as duas VPCs. Para obter mais informações, consulte [VPC Peering](#) no Guia do usuário da Amazon Virtual Private Cloud.
- Você deve criar o grupo de segurança que deseja anexar ao nó principal na VPC do cluster. Depois que as duas VPCs tiverem sido emparelhadas, a vinculação cruzada entre os grupos de segurança do lado do banco de dados e do cluster ficará disponível. Para obter mais informações, consulte [Regras do Grupo de Segurança](#) no Guia do usuário da Amazon Virtual Private Cloud.

Configurando a criptografia TLS entre `slurmdbd` e o servidor de banco de dados

Com a configuração de Slurm contabilidade padrão fornecida, AWS ParallelCluster `slurmdbd` estabelece uma conexão criptografada TLS com o servidor de banco de dados, se o servidor oferecer suporte à criptografia TLS. AWS serviços de banco de dados, como o Amazon RDS, Amazon Aurora oferecem suporte à criptografia TLS por padrão.

Você pode exigir conexões seguras no lado do servidor definindo o parâmetro `requires_secure_transport` no servidor do banco de dados. Isso é configurado no CloudFormation modelo fornecido.

Seguindo as práticas recomendadas de segurança, recomendamos também habilitar a verificação da identidade do servidor no cliente do `slurmdbd`. Para fazer isso, configure [StorageParameters](#). no `slurmdbd.conf`. Carregue o certificado CA do servidor no nó principal do cluster. Em seguida, defina a opção [SSL_CA](#) de `StorageParameters` em `slurmdbd.conf` para o caminho do certificado CA do servidor no nó principal. Isso permite a verificação da identidade do servidor no lado de `slurmdbd`. Depois de fazer essas alterações, reinicie o serviço `slurmdbd` para restabelecer a conectividade com o servidor de banco de dados com a verificação de identidade ativada.

Atualização das credenciais do banco de dados

Para atualizar os valores [de Database/Username](#) ou [PasswordSecretArn](#), você deve primeiro interromper a frota computacional. Suponha que o valor secreto armazenado no AWS Secrets Manager segredo seja alterado e seu ARN não seja alterado. Nessa situação, o cluster não atualiza automaticamente a senha do banco de dados para o novo valor. Para atualizar o cluster para o novo valor secreto, execute o comando a seguir no nó principal.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

Warning

Para evitar a perda de dados contábeis, recomendamos que você altere a senha do banco de dados somente quando a frota de computação for interrompida.

Monitoramento de banco de dados

Recomendamos que você ative os recursos de monitoramento dos serviços de AWS banco de dados. Para obter mais informações, consulte a documentação de [Monitoramento do Amazon RDS](#) ou de [Monitoramento do Amazon Aurora](#).

Personalização de configuração do Slurm

A partir do AWS ParallelCluster versão 3.6.0, você pode personalizar a configuração do Slurm `slurm.conf` em uma configuração de cluster do AWS ParallelCluster.

Na configuração do cluster, você pode personalizar os parâmetros de configuração do Slurm usando as seguintes definições de configuração do cluster:

- Personalize os parâmetros do Slurm para todo o cluster usando [SlurmSettings](#) / [CustomSlurmSettings](#) ou o parâmetro [CustomSlurmSettingsIncludeFile](#). O AWS ParallelCluster falhará se você especificar ambos.
- Personalize os parâmetros do Slurm de uma fila usando [SlurmQueues](#) / [CustomSlurmSettings](#) (mapeado para partições Slurm).
- Personalize os parâmetros do Slurm de um recurso de computação usando [SlurmQueues](#) / [ComputeResources](#) / [CustomSlurmSettings](#) (mapeado para nós do Slurm).

Considerações de limites de personalização de configuração do Slurm ao usar o AWS ParallelCluster

- Para as configurações `CustomSlurmSettings` e `CustomSlurmSettingsIncludeFile`, você só pode especificar e atualizar os parâmetros do `slurm.conf` incluídos na [Slurmversão](#) compatível com a versão do AWS ParallelCluster que você está usando para configurar um cluster.
- Se você especificar configurações personalizadas do Slurm em qualquer um dos parâmetros `CustomSlurmSettings`, AWS ParallelCluster executará verificações de validação e evitará definir ou atualizar parâmetros de configuração do Slurm que entrem em conflito com a lógica do AWS ParallelCluster. Os parâmetros de configuração do Slurm com os quais se sabe que entram em conflito com o AWS ParallelCluster são identificados nas listas de negação. As listas de negação podem mudar em versões futuras do AWS ParallelCluster se outros recursos do Slurm forem adicionados. Para obter mais informações, consulte [Parâmetros de configuração do Slurm listados como negados para CustomSlurmSettings](#).
- O AWS ParallelCluster só verifica se um parâmetro está em uma lista de negação. O AWS ParallelCluster não valida a sintaxe ou a semântica do parâmetro de configuração do Slurm personalizada por você. Você é responsável por validar seus parâmetros de configuração personalizados do Slurm. Parâmetros de configuração personalizados do Slurm inválidos podem causar falhas no daemon do Slurm que podem levar a falhas na criação e atualização do cluster.
- Se você especificar configurações personalizadas do Slurm em `CustomSlurmSettingsIncludeFile`, o AWS ParallelCluster não executará nenhuma validação.
- Você pode atualizar `CustomSlurmSettings` e `CustomSlurmSettingsIncludeFile` sem parar e iniciar a frota de computação. Nesse caso, AWS ParallelCluster reinicia o daemon do `slurmctld` e executa o comando `scontrol reconfigure`.

Alguns parâmetros de configuração Slurm podem exigir operações diferentes antes que uma alteração seja registrada em todo o cluster. Por exemplo, eles podem exigir a reinicialização de todos os daemons no cluster. Você é responsável por verificar se as operações do AWS ParallelCluster são suficientes para propagar suas definições de parâmetros personalizados de configuração do Slurm durante as atualizações. Se você achar que as operações do AWS ParallelCluster não são suficientes, é sua responsabilidade fornecer as ações adicionais necessárias para propagar as configurações atualizadas, conforme recomendado na [documentação do Slurm](#).

Parâmetros de configuração do Slurm listados como negados para **CustomSlurmSettings**

As tabelas a seguir listam os parâmetros com as versões do AWS ParallelCluster que negam seu uso, começando com a versão 3.6.0. CustomSlurmSettings não é compatível com versões do AWS ParallelCluster anteriores à versão 3.6.0.

Parâmetros listados na lista de negação no nível do cluster:

Parâmetro Slurm	Lista de negação nas versões do AWS ParallelCluster
CommunicationParameters	3.6.0
Epilog	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

Parâmetro Slurm	Lista de negação nas versões do AWS ParallelCluster
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

Parâmetros listados na lista de negação no nível do cluster quando a [integração contábil nativa do Slurm](#) está configurada na configuração do cluster:

Parâmetro Slurm	Lista de negação nas versões do AWS ParallelCluster
AccountingStorageType	3.6.0
AccountingStorageHost	3.6.0
AccountingStoragePort	3.6.0
AccountingStorageUser	3.6.0
JobAcctGatherType	3.6.0

Parâmetros listados na lista de negação no nível da fila (partição) para filas gerenciadas pelo AWS ParallelCluster:

Parâmetro Slurm	Lista de negação nas versões do AWS ParallelCluster
Nodes	3.6.0
PartitionName	3.6.0
ResumeTimeout	3.6.0
State	3.6.0
SuspendTime	3.6.0

Parâmetros listados na lista de negação no nível do recurso de computação (nó) para recursos de computação gerenciados pelo AWS ParallelCluster:

Parâmetro Slurm	Lista de negação na versão do AWS ParallelCluster e em versões posteriores
CPUs	3.6.0
Recursos	3.6.0
Gres	3.6.0
NodeAddr	3.6.0
NodeHostname	3.6.0
NodeName	3.6.0
Weight	3.7.0

Slurm **prolog** e **epilog**

A partir da versão 3.6.0 do AWS ParallelCluster, a configuração Slurm implantada com o AWS ParallelCluster inclui os parâmetros de configuração Prolog 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
```

Para obter mais informações, consulte o [Guia do Prolog e do Epilog](#) na documentação do Slurm.

O AWS ParallelCluster inclui os seguintes scripts prolog e epilog:

- `90_plcluster_health_check_manager` (na pasta Prolog)
- `90_pcluster_noop` (na pasta Epilog)

Note

Tanto a pasta Prolog quanto a Epilog devem ter pelo menos um arquivo.

Você pode usar seus próprios scripts prolog e epilog personalizados, adicionando-os às pastas Prolog e Epilog correspondentes.

Warning

Slurm executa todos os scripts nas pastas, em ordem alfabética inversa.

A duração do tempo de execução dos scripts prolog e epilog afeta o tempo necessário para executar um trabalho. Atualize a configuração `BatchStartTimeout` ao executar vários scripts ou scripts prolog de longa execução. O padrão é 3 minutos.

Se você estiver usando scripts prolog e epilog personalizados, localize os scripts nas pastas Prolog e Epilog correspondentes. Recomendamos que você mantenha o script `90_plcluster_health_check_manager` que é executado antes de cada script personalizado. Para obter mais informações, consulte [Personalização de configuração do Slurm](#).

Tamanho e atualização da capacidade do cluster

A capacidade do cluster é definida pelo número de nós de computação que o cluster pode escalar. Os nós de computação são apoiados por instâncias do EC2 definidas nos recursos computacionais

da AWS ParallelCluster configuração (Scheduling/SlurmQueues/[ComputeResources](#)) e são organizados em filas (Scheduling/[SlurmQueues](#)) que mapeiam 1:1 para partições. Slurm

Em um recurso de computação, é possível configurar o número mínimo de nós de computação (instâncias) que sempre devem ser mantidos em execução no cluster (MinCount) e o número máximo de instâncias para as quais o recurso computacional pode ser escalado (MaxCount3).

No momento da criação do cluster, ou após uma atualização do cluster AWS ParallelCluster, executa quantas instâncias do EC2 forem configuradas MinCount para cada recurso computacional (Scheduling/SlurmQueues/ [ComputeResources](#)) definido no cluster. As instâncias lançadas para cobrir a quantidade mínima de nós para os recursos computacionais no cluster são chamadas de nós estáticos. Depois de iniciados, os nós estáticos devem ser persistentes no cluster e não são encerrados pelo sistema, a menos que ocorra um evento ou condição específica. Esses eventos incluem, por exemplo, a falha das verificações de integridade do EC2 e a alteração do status do nó Slurm para DRAIN Slurm ou DOWN.

As instâncias do EC2, na faixa de **1** até '**MaxCount - MinCount**' (**MaxCount** menos) **MinCount**), lançadas sob demanda para lidar com o aumento da carga do cluster, são chamadas de nós dinâmicos. Sua natureza é efêmera, eles são iniciados para atender a trabalhos pendentes e são encerrados quando permanecem ociosos por um período de tempo definido Scheduling/SlurmSettings/[ScaledownIdleTime](#) na configuração do cluster (padrão: 10 minutos).

Os nós estáticos e os nós dinâmicos estão em conformidade com o seguinte esquema de nomenclatura:

- Nós estáticos <Queue/Name>-st-<ComputeResource/Name>-<num> onde <num> = 1..ComputeResource/MinCount
- Nós dinâmicos <Queue/Name>-dy-<ComputeResource/Name>-<num> onde <num> = 1.. (ComputeResource/MaxCount - ComputeResource/MinCount)

Por exemplo, dada a seguinte AWS ParallelCluster configuração:

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
```



```
Instances:
  - InstanceType: c5.xlarge
  MinCount: 100
  MaxCount: 150
```

Os seguintes nós serão definidos em 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 um recurso computacional o tiver `MinCount == MaxCount`, todos os nós de computação correspondentes serão estáticos e todas as instâncias serão iniciadas no momento da criação/atualização do cluster e mantidas em funcionamento. Por exemplo: .

```
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]
```

Atualização da capacidade do cluster

A atualização da capacidade do cluster inclui a adição ou remoção de filas, recursos computacionais ou a alteração `MinCount/MaxCount` de um recurso computacional. A partir da AWS ParallelCluster

versão 3.9.0, reduzir o tamanho de uma fila exige que a frota de computação seja interrompida ou [QueueUpdateStrategy](#) configurada como TERMINATE antes que uma atualização do cluster ocorra. Não é necessário interromper a frota de computação ou configurar [QueueUpdateStrategy](#) para TERMINATE quando:

- Adicionando novas filas ao Scheduling/ [SlurmQueues](#)
- Adicionar novos recursos computacionais Scheduling/SlurmQueues/[ComputeResources](#) a uma fila
- Aumentando a [MaxCount](#) quantidade de um recurso computacional
- Aumento MinCount de um recurso computacional e aumento MaxCount do mesmo recurso computacional em pelo menos a mesma quantidade

Considerações e limitações

Esta seção tem como objetivo descrever quaisquer fatores, restrições ou limitações importantes que devem ser levados em consideração ao redimensionar a capacidade do cluster.

- Ao remover uma fila de Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues todos os nós de computação com nome <Queue/Name> - *, tanto estático quanto dinâmico, ela será removida da Slurm configuração e as instâncias EC2 correspondentes serão encerradas.
- Ao remover um recurso computacional Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues-ComputeResources> ComputeResources de uma fila, todos os nós de computação com nome <Queue/Name> - * - <ComputeResource/Name> - *, estático e dinâmico, serão removidos da Slurm configuração e as instâncias EC2 correspondentes serão encerradas.

Ao alterar o MinCount parâmetro de um recurso computacional, podemos distinguir dois cenários diferentes: se MaxCount é mantido igual a MinCount (somente capacidade estática) e se MaxCount é maior que MinCount (capacidade estática e dinâmica mista).

Alterações de capacidade somente com nós estáticos

- Se `MinCount == MaxCount`, ao aumentar `MinCount` (e `MaxCount`), o cluster for configurado estendendo o número de nós estáticos até o novo valor de, `MinCount <Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` e o sistema continuar tentando iniciar instâncias do EC2 para atender à nova capacidade estática necessária.
- Se `MinCount == MaxCount`, ao diminuir `MinCount` (e `MaxCount`) da quantidade `N`, o cluster for configurado removendo os últimos `N` nós estáticos `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`] e o sistema encerrará as instâncias EC2 correspondentes.
- Estado inicial `MinCount = MaxCount = 100`

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

- Atualizar `-30` em `MinCount` e `MaxCount`: `MinCount = MaxCount = 70`

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

Alterações de capacidade com nós mistos

Se `MinCount < MaxCount`, `MinCount` ao aumentar em uma quantidade `N` (supondo que `MaxCount` será mantido inalterado), o cluster for configurado estendendo o número de nós estáticos até o novo valor de `MinCount` (`old_MinCount + N`): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` e o sistema continuará tentando iniciar instâncias do EC2 para atender à nova capacidade estática necessária. Além disso, para honrar a `MaxCount` capacidade do recurso computacional, a configuração do cluster é atualizada removendo os últimos `N` nós dinâmicos: `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` e o sistema encerrará as instâncias EC2 correspondentes.

- Estado inicial: 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]
```

- Atualize +30 para MinCount : MinCount = 130 (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 $\text{MinCount} < \text{MaxCount}$, ao aumentar MinCount e MaxCount com a mesma quantidade N, o cluster for configurado estendendo o número de nós estáticos para o novo valor de MinCount ($\text{old_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` e o sistema continuará tentando iniciar instâncias do EC2 para atender à nova capacidade estática necessária. Além disso, nenhuma alteração será feita no número de nós dinâmicos para honrar o novo

MaxCount value.

- Estado inicial: 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]
```

- Atualize +30 para 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 $\text{MinCount} < \text{MaxCount}$, ao diminuir a quantidade N (MinCount supondo que MaxCount será mantida inalterada), o cluster será configurado removendo os últimos N nós estáticos dos nós estáticos `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>` e o sistema encerrará as instâncias EC2 correspondentes. Além disso, para honrar a MaxCount capacidade do recurso computacional, a configuração do cluster é atualizada ampliando o número de nós dinâmicos para preencher a lacuna. $\text{MaxCount} - \text{new_MinCount}$: `<Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]` Nesse caso, como esses são nós dinâmicos, nenhuma nova instância do EC2 será iniciada, a menos que o agendador tenha trabalhos pendentes nos novos nós.

- Estado inicial: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

```
$ 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]
```

- Atualização -30 ativada MinCount : $\text{MinCount} = 70$ ($\text{MaxCount} = 120$)

```
$ 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 $\text{MinCount} < \text{MaxCount}$, ao diminuir MinCount e MaxCount com a mesma quantidade N , o cluster for configurado removendo os últimos N nós estáticos `<Queue/Name>-st-`

<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>] e o sistema encerrará as instâncias EC2 correspondentes.

Além disso, nenhuma alteração será feita no número de nós dinâmicos para honrar o novo MaxCount valor.

- Estado inicial: 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]
```

- Atualização -30 ativada 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 $MinCount < MaxCount$, ao diminuir a quantidade N ($MaxCount$ supondo que $MinCount$ será mantida inalterada), o cluster for configurado removendo os últimos N nós dinâmicos <Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>] e o sistema encerrará as instâncias EC2 correspondentes no caso de elas estarem em execução. Nenhum impacto é esperado nos nós estáticos.

- Estado inicial: 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]
```

- Atualização -30 ativada 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]
```

Impactos nos empregos

Em todos os casos em que os nós são removidos e as instâncias do EC2 encerradas, um trabalho em lote executado nos nós removidos será colocado novamente na fila, a menos que não haja outros nós que satisfaçam os requisitos do trabalho. Nesse último caso, o trabalho falhará com o status `NODE_FAIL` e desaparecerá da fila; se for o caso, ele precisará ser reenviado manualmente.

Se você planeja realizar uma atualização de redimensionamento do cluster, pode impedir que os trabalhos sejam executados nos nós que serão removidos durante a atualização planejada. Isso é possível configurando os nós a serem removidos na manutenção. Esteja ciente de que a configuração de um nó em manutenção não afetaria as tarefas que eventualmente já estão sendo executadas no nó.

Suponha que, com a atualização planejada de redimensionamento do cluster, você remova o nó `queue-st-computeresource-[9-10]`. Você pode criar uma Slurm reserva com o seguinte comando

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queue-st-
computeresource-[9-10]
```

Isso criará uma Slurm reserva nomeada `maint_for_update` nos nós `queue-st-computeresource-[9-10]`. A partir do momento em que a reserva é criada, nenhuma outra tarefa pode ser executada nos nós `queue-st-computeresource-[9-10]`. Esteja ciente de que a reserva não impedirá que os trabalhos sejam eventualmente alocados nos nós `queue-st-computeresource-[9-10]`.

Após a atualização do redimensionamento do cluster, se a Slurm reserva tiver sido definida somente nos nós que foram removidos durante a atualização do redimensionamento, a reserva

de manutenção será excluída automaticamente. Se, em vez disso, você tiver criado uma Slurm reserva nos nós que ainda estão presentes após a atualização do redimensionamento do cluster, talvez queiramos remover a reserva de manutenção nos nós após a execução da atualização de redimensionamento, usando o seguinte comando

```
sudo -i scontrol delete ReservationName=maint_for_update
```

Para obter detalhes adicionais sobre a Slurm reserva, consulte o documento oficial do SchedMD [aqui](#).

Processo de atualização do cluster sobre mudanças de capacidade

Após uma alteração na configuração do agendador, as seguintes etapas são executadas durante o processo de atualização do cluster:

- Pare AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- Gere a configuração de Slurm partições atualizada a partir da AWS ParallelCluster configuração
- Reiniciar `slurmctld` (feito por meio da receita do serviço Chef)
- Verifique o `slurmctld` status (`systemctl is-active --quiet slurmctld.service`)
- Recarregar configuração Slurm (`scontrol reconfigure`)
- Iniciar `clustermgtd` (`supervisorctl start clustermgtd`)

AWS Batch (**awsbatch**)

Para obter mais informações sobre o AWS Batch, consulte [AWS Batch](#). Para obter a documentação, consulte o [Guia do usuário do AWS Batch](#).

Comandos da CLI do AWS ParallelCluster para o AWS Batch

Quando você usa o programador `awsbatch`, os comandos da CLI do AWS ParallelCluster para o AWS Batch são instalados automaticamente no nó principal do AWS ParallelCluster. A CLI usa operações de API AWS Batch e permite as seguintes operações:

- Enviar e gerenciar trabalhos.
- Monitorar trabalhos, filas e hosts.
- Espelhar comandos do programador tradicional.

⚠ Important

O AWS ParallelCluster não é compatível com trabalhos de GPU para AWS Batch. Para obter mais informações, consulte [Trabalhos de GPU](#).

Esta CLI é distribuída como um pacote separado. Para obter mais informações, consulte [Compatibilidade com programadores](#).

Tópicos

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)
- [awsbqueues](#)
- [awsbhosts](#)

awsbsub

Envia trabalhos para a fila de trabalhos do 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

O AWS ParallelCluster não é compatível com trabalhos de GPU para AWS Batch. Para obter mais informações, consulte [Trabalhos de GPU](#).

Argumentos posicionais

command

Envia o trabalho (o comando especificado deve estar disponível nas instâncias de computação) ou o nome do arquivo a ser transferido. Consulte também `--command-file`.

arguments

(Opcional) Especifica argumentos para o comando ou arquivo de comando.

Argumentos nomeados

-jn *JOB_NAME*, --job-name *JOB_NAME*

Nomeia a tarefa. O primeiro caractere deve ser uma letra ou um número. O nome do trabalho pode conter letras (minúsculas e maiúsculas), números, hifens e sublinhados, e ter até 128 caracteres de comprimento.

-c *CLUSTER*, --cluster *CLUSTER*

Especifica o cluster a ser usado.

-cf, --command-file

Indica que o comando é um arquivo a ser transferido para as instâncias de computação.

Padrão: falso

-w *WORKING_DIR*, --working-dir *WORKING_DIR*

Especifica a pasta a ser usada como diretório de trabalho da tarefa. Se um diretório de trabalho não for especificado, o trabalho será executado na subpasta `job-<AWS_BATCH_JOB_ID>` do diretório inicial do usuário. Você pode usar esse parâmetro ou o parâmetro `--parent-working-dir`.

-pw *PARENT_WORKING_DIR*, --parent-working-dir *PARENT_WORKING_DIR*

Especifica a pasta pai do diretório de trabalho da tarefa. Se um diretório de trabalho pai não for especificado, o padrão será o diretório inicial do usuário. Uma subpasta chamada `job-<AWS_BATCH_JOB_ID>` será criada no diretório de trabalho pai. Você pode usar esse parâmetro ou o parâmetro `--working-dir`.

-if *INPUT_FILE*, --input-file *INPUT_FILE*

Especifica o arquivo a ser transferido para as instâncias de computação, no diretório de trabalho do trabalho. Você pode especificar vários parâmetros de arquivo de entrada.

-p *VCPUS*, --vcpus *VCPUS*

Especifica o número de vCPUs a serem reservadas para o contêiner. Quando usado em conjunto com `-nodes`, ele identifica o número de vCPUs para cada nó.

Padrão: 1

-m *MEMORY*, --memory *MEMORY*

Especifica o limite rígido de memória (em MiB) a ser fornecido para a tarefa. Se o trabalho tentar exceder o limite de memória especificado aqui, ele será encerrado.

Padrão: 128

-e *ENV*, --env *ENV*

Especifica uma lista separada por vírgulas dos nomes das variáveis de ambiente a serem exportadas para o ambiente da tarefa. Para exportar todas as variáveis de ambiente, especifique "all". Observe que uma lista de variáveis de ambiente "all" não inclui as listadas no parâmetro `-env-blacklist`, nem variáveis que começam com os prefixos `PCLUSTER_*` e `AWS_*`.

-eb *ENV_DENYLIST*, --env-blacklist *ENV_DENYLIST*

Especifica uma lista separada por vírgulas dos nomes das variáveis de ambiente a não serem exportadas para o ambiente da tarefa. Por padrão, `HOME`, `PWD`, `USER`, `PATH`, `LD_LIBRARY_PATH`, `TERM` e `TERMCAP` não são exportadas.

-r *RETRY_ATTEMPTS*, --retry-attempts *RETRY_ATTEMPTS*

Especifica o número de vezes que um trabalho será movido para o status `RUNNABLE`. Você pode especificar entre 1 e 10 tentativas. Se o valor de tentativas for maior que 1, o trabalho será repetido se falhar, até ser movido para um status `RUNNABLE` esse número especificado de vezes.

Padrão: 1

-t *TIMEOUT*, --timeout *TIMEOUT*

Especifica duração em segundos (contando a partir do timestamp `startedAt` da tentativa de trabalho) após a qual o AWS Batch encerrará seu trabalho caso não tenha sido concluído. O valor de tempo limite deve ser pelo menos 60 segundos.

-n *NODES*, --nodes *NODES*

Especifica o número de nós a serem reservados para a tarefa. Especifique um valor para esse parâmetro a fim de habilitar o envio paralelo com vários nós.

Note

Quando o parâmetro [Scheduler](#) / [AwsBatchQueues](#) / [CapacityType](#) é definido como SPOT, trabalhos paralelos de vários nós não são suportados. Além disso, deve haver uma função `AWSServiceRoleForEC2Spot` vinculada ao serviço em sua conta. Você pode criar essa função com o comando de AWS CLI a seguir:

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

Para obter mais informações, consulte [Função vinculada ao serviço para solicitações de instâncias spot](#) no Guia do usuário do Amazon EC2 para instâncias do Linux

-a *ARRAY_SIZE*, --array-size *ARRAY_SIZE*

Indica o tamanho da matriz. Você pode especificar um valor entre 2 e 10.000. Se você especificar propriedades de matriz para uma tarefa, ela se torna uma tarefa de matriz.

-d *DEPENDS_ON*, --depends-on *DEPENDS_ON*

Especifica uma lista de dependências separadas por ponto-e-vírgula para uma tarefa. Uma tarefa pode depender de, no máximo, outras 20 tarefas. Você pode especificar um tipo de dependência SEQUENTIAL sem especificar um ID de trabalho para trabalhos de matriz. Uma dependência sequencial permite que cada tarefa filho da matriz seja concluída sequencialmente, a partir do índice 0. Você também pode especificar uma dependência do tipo N_TO_N com um ID de tarefa para tarefas de matriz. Uma dependência N_TO_N significa que cada índice filho dessa tarefa precisa aguardar que o índice filho correspondente de cada dependência seja concluído antes de poder começar. A sintaxe desse parâmetro é "jobId=<string>,type=<string>;...".

awsbstat

Mostra as tarefas que são enviadas na fila de tarefas do cluster.

```
awsbstat [-h] [-c CLUSTER] [-s STATUS] [-e] [-d] [job_ids [job_ids ...]]
```

Argumentos posicionais

job_ids

Especifica a lista separada por espaços de IDs de tarefas a serem exibidas na saída. Se o trabalho for uma matriz de trabalhos, todos os trabalhos filho são exibidos. Se uma única tarefa for solicitada, seus detalhes são exibidos.

Argumentos nomeados

-c *CLUSTER*, --cluster *CLUSTER*

Indica o cluster a ser usado.

-s *STATUS*, --status *STATUS*

Especifica uma lista separada por vírgulas de status de tarefa a ser incluída. O status de tarefa padrão é "ativo". Os valores aceitos são: SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING, SUCCEEDED, FAILED e ALL.

Padrão: "SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING"

-e, --expand-children

Expande tarefas com filhos (matriz e paralelo com vários nós).

Padrão: falso

-d, --details

Mostra os detalhes da tarefa.

Padrão: falso

awsbout

Mostra a saída de uma tarefa específica.

```
awsbout [-h] [-c CLUSTER] [-hd HEAD] [-t TAIL] [-s] [-sp STREAM_PERIOD] job_id
```

Argumentos posicionais

job_id

Especifica o ID da tarefa.

Argumentos nomeados

-c *CLUSTER*, --cluster *CLUSTER*

Indica o cluster a ser usado.

-hd *HEAD*, --head *HEAD*

Obtém as primeiras linhas de *HEAD* (*Título*) da saída do trabalho.

-t *TAIL*, --tail *TAIL*

Obtém as últimas linhas <finais> da saída da tarefa.

-s, --stream

Obtém a saída da tarefa e aguarda a saída adicional que será produzida. Esse argumento pode ser usado em conjunto com `-tail` para iniciar a partir das últimas linhas <finais> da saída da tarefa.

Padrão: falso

-sp *STREAM_PERIOD*, --stream-period *STREAM_PERIOD*

Define o período de streaming.

Padrão: 5

awsbkill

Cancela ou encerra tarefas enviadas no cluster.

```
awsbkill [-h] [-c CLUSTER] [-r REASON] job_ids [job_ids ... ]
```

Argumentos posicionais

job_ids

Especifica a lista separada por espaços de IDs de tarefas a serem canceladas ou encerradas.

Argumentos nomeados

-c *CLUSTER*, --cluster *CLUSTER*

Indica o nome do cluster a ser usado.

-r *REASON*, --reason *REASON*

Indica a mensagem que será anexada a uma tarefa, explicando o motivo do cancelamento.

Padrão: "Terminated by the user"

awsbqueues

Mostra a fila de tarefas associada ao cluster.

```
awsbqueues [-h] [-c CLUSTER] [-d] [job_queues [job_queues ... ]]
```

Argumentos posicionais

job_queues

Especifica a lista separada por espaços de nomes de fila a ser exibida. Se uma única fila for solicitada, seus detalhes são exibidos.

Argumentos nomeados

-c *CLUSTER*, --cluster *CLUSTER*

Especifica o nome do cluster a ser usado.

-d, --details

Indica se deve mostrar os detalhes das filas.

Padrão: falso

awsbhosts

Mostra os hosts que pertencem ao ambiente de computação do cluster.

```
awsbhosts [-h] [-c CLUSTER] [-d] [instance_ids [instance_ids ... ]]
```

Argumentos posicionais

instance_ids

Especifica uma lista separada por espaços de IDs de instâncias. Se uma única instância for solicitada, seus detalhes são exibidos.

Argumentos nomeados

-c *CLUSTER*, --cluster *CLUSTER*

Especifica o nome do cluster a ser usado.

-d, --details

Indica se deve mostrar os detalhes dos hosts.

Padrão: falso

Armazenamento compartilhado

AWS ParallelCluster suporta o uso do [Amazon EBS](#), [FSx para ONTAP](#), e volumes de armazenamento compartilhado [FSx para OpenZFS](#), [Amazon EFS](#) e sistemas de armazenamento de arquivo compartilhado [FSx para Lustre](#) ou [Caches de arquivos](#). Recomendamos que você siga a orientação do [pilar de confiabilidade da estrutura bem arquitetada AWS](#) e faça backup de seus volumes e sistemas de arquivos.

Selecione um sistema de armazenamento que atenda aos requisitos de I/O do seu aplicativo HPC. Você pode otimizar cada sistema de arquivos com base em seu caso de uso específico. Para obter mais informações, consulte [visão geral sobre opções de armazenamento](#).

Os volumes do Amazon EBS são anexados ao nó principal e compartilhados com os nós de computação por meio do NFS. Essa opção pode ser econômica, mas o desempenho depende dos recursos do nó principal à medida que as necessidades de armazenamento aumentam. Isso pode se tornar um gargalo à medida que mais nós de computação são adicionados ao cluster e a demanda de taxa de throughput aumenta.

Os sistemas de arquivos do Amazon EFS escalam conforme as necessidades de armazenamento mudam. Você pode configurar esses sistemas de arquivos para uma variedade de casos de uso. Use

os sistemas de arquivos Amazon EFS para executar aplicativos paralelizados e sensíveis à latência em seu cluster.

Os sistemas de arquivos FSx para Lustre podem processar grandes conjuntos de dados com taxa de throughput de até centenas de gigabytes por segundo, milhões de IOPS e latências inferiores a um milissegundo. Use os sistemas de arquivos FSx para Lustre para ambientes computacionais que exigem alto desempenho.

Em [Seção SharedStorage](#), você pode definir armazenamento externo ou gerenciado AWS ParallelCluster:

- O armazenamento externo se refere a um volume ou sistema de arquivos existente que você gerencia. AWS ParallelCluster não cria nem exclui esse armazenamento.
- O armazenamento gerenciado se refere a um volume ou sistema de arquivos criado e excluído por AWS ParallelCluster.

Armazenamento externo

Você pode configurar AWS ParallelCluster para anexar armazenamento externo ao cluster quando o cluster é criado ou atualizado. Da mesma forma, você pode configurá-lo para separar o armazenamento externo do cluster quando o cluster for excluído ou atualizado. Seus dados são preservados e você pode usá-los para armazenamento compartilhado permanente de longo prazo fora do ciclo de vida do cluster.

Note

As versões AWS ParallelCluster anteriores à 3.8 não permitem a montagem de sistemas de arquivos gerenciados externamente no `/home`. A partir da versão 3.8, AWS ParallelCluster permite que você use `/home` como ponto de montagem para um sistema de arquivos gerenciado externo. Você pode montar um sistema de arquivos gerenciado externamente `/home` especificando `/home` o valor do [MountDir](#) parâmetro abaixo do [Seção SharedStorage](#).

O Amazon File Cache não é adequado para uso como `/home` diretório do sistema e, portanto, não é suportado no momento para montagem `/home`.

Ao especificar um `/home` diretório abaixo, [Seção SharedStorage](#) a opção de [SharedStorageType](#) configuração será substituída, o que significa que as configurações abaixo [Seção SharedStorage](#) serão usadas em seu lugar.

Ao montar um sistema de arquivos externo no /home diretório, AWS ParallelCluster copia o /home conteúdo do nó principal para o sistema de arquivos externo, sem sobrescrever os arquivos existentes no armazenamento externo. Isso inclui transferir a chave SSH do cluster para o usuário padrão, se ela estiver ausente no sistema de arquivos externo. Para obter mais informações, consulte [Considerações sobre armazenamento compartilhado do AWS ParallelCluster](#).

armazenamento gerenciado AWS ParallelCluster

Por padrão, o armazenamento gerenciado do AWS ParallelCluster depende do ciclo de vida do cluster na configuração. Por padrão, o parâmetro de configuração SharedStorage DeletionPolicy é definido como Delete.

Por padrão, um sistema de arquivos ou volume gerenciado do AWS ParallelCluster e seus dados são excluídos se uma das afirmações a seguir for verdadeira.

- Se você excluir o cluster.
- Se você alterar a configuração do armazenamento compartilhado gerenciado do Name.
- Se você remover o armazenamento compartilhado gerenciado da configuração.

Defina DeletionPolicy como Retain para manter seu sistema de arquivos ou volume compartilhado gerenciado e seus dados. Recomendamos que você faça backup de seus dados regularmente para evitar a perda de dados. Você pode usar o [AWS Backup](#) para gerenciar centralmente os backups de todas as suas opções de armazenamento.

Você pode remover a dependência do ciclo de vida com as configurações. Para obter mais informações, consulte [Converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo](#).

Para obter mais informações, sobre cotas de armazenamento compartilhado, consulte [Cotas para armazenamento compartilhado](#).

Para obter mais informações sobre armazenamento compartilhado e mudança para novas versões do AWS ParallelCluster, consulte [Práticas recomendadas: mover um cluster para uma nova versão de patch ou secundária do AWS ParallelCluster](#).

Você pode configurar o AWS ParallelCluster para anexar armazenamento externo ao cluster quando o cluster é criado ou atualizado. Da mesma forma, você pode configurá-lo para separar o

armazenamento externo do cluster quando o cluster for excluído ou atualizado. Seus dados são preservados e você pode usá-los para soluções de armazenamento compartilhado permanente de longo prazo que são independentes do ciclo de vida do cluster.

Por padrão, o armazenamento gerenciado depende do ciclo de vida do cluster. Você pode remover a dependência do ciclo de vida com as configurações descritas em [Converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo](#).

Com configurações específicas, você pode otimizar cada uma das soluções de armazenamento suportadas para seus casos de uso.

Para cotas de armazenamento compartilhado, consulte [Cotas para armazenamento compartilhado](#).

Para obter mais informações sobre armazenamento compartilhado e mudança para novas versões do AWS ParallelCluster, consulte [Práticas recomendadas: mover um cluster para uma nova versão de patch ou secundária do AWS ParallelCluster](#).

Tópicos

- [Configurar o armazenamento compartilhado](#)
- [Trabalhando com armazenamento compartilhado no AWS ParallelCluster](#)
- [Cotas para armazenamento compartilhado](#)

Configurar o armazenamento compartilhado

Saiba mais sobre as configurações que podem ser usadas para definir o armazenamento compartilhado do cluster.

Tópicos

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx para Lustre](#)
- [Configurar armazenamento compartilhado do FSx para ONTAP, FSx para OpenZFS e File Cache](#)

Amazon Elastic Block Store

Para usar um volume externo existente do Amazon EBS para armazenamento permanente de longo prazo que seja independente do ciclo de vida do cluster, especifique [EbsSettings](#) / [VolumeId](#).

Se você não especificar [VolumeId](#), por padrão, o AWS ParallelCluster cria um volume gerenciado do EBS a partir de [EbsSettings](#) quando seu cluster é criado. O AWS ParallelCluster também exclui o volume e os dados quando o cluster é excluído ou o volume é removido da configuração do cluster.

Para um volume do EBS gerenciado pelo AWS ParallelCluster, você pode usar [EbsSettings](#) / [DeletionPolicy](#) para instruir o AWS ParallelCluster a Delete, Retain ou Snapshot no volume quando o cluster for excluído ou quando o volume for removido da configuração do cluster. Por padrão, DeletionPolicy é definido como Delete.

Warning

Para armazenamento compartilhado gerenciado pelo AWS ParallelCluster, DeletionPolicy é definido como Delete por padrão.

Isso significa que, se uma das afirmações a seguir for verdadeira, um volume gerenciado e seus dados serão excluídos:

- Se você excluir o cluster.
- Se você alterar a configuração de armazenamento compartilhado gerenciado [SharedStorage](#) / [Name](#).
- Se você remover o armazenamento compartilhado gerenciado da configuração.

Recomendamos que você faça backup de seus dados regularmente para evitar a perda de dados. Para obter mais informações sobre snapshots do Amazon EBS, consulte [Amazon EBS snapshots](#) no Guia do usuário do Amazon Elastic Compute Cloud para Instâncias do Linux. Para saber como gerenciar backups de dados em Serviços da AWS, consulte [AWSBackup](#) no AWS Backup Guia do desenvolvedor.

Amazon Elastic File System

Para usar um sistema de arquivos externo existente do Amazon EFS para armazenamento permanente de longo prazo fora do ciclo de vida do cluster, especifique [EfsSettings](#) / [FileSystemId](#), por padrão, AWS ParallelCluster cria um sistema de arquivos gerenciado do Amazon EFS a partir de [EfsSettings](#) no momento em que cria o cluster. AWS ParallelCluster também exclui o sistema de arquivos e os dados quando o cluster é excluído ou quando o sistema de arquivos é removido da configuração do cluster.

Para um sistema de arquivos do Amazon EFS gerenciado pelo AWS ParallelCluster, você pode usar [EfsSettings](#) / [DeletionPolicy](#) para instruir o AWS ParallelCluster a Delete ou Retain quando o cluster for excluído ou quando o sistema de arquivos for removido da configuração do cluster. Por padrão, DeletionPolicy é definido como Delete.

Warning

Para armazenamento compartilhado gerenciado pelo AWS ParallelCluster, DeletionPolicy é definido como Delete por padrão.

Isso significa que, se uma das afirmações a seguir for verdadeira, um sistema de arquivos gerenciado e seus dados serão excluídos:

- Se você excluir o cluster.
- Se você alterar a configuração de armazenamento compartilhado gerenciado [SharedStorage](#) / [Name](#).
- Se você remover o armazenamento compartilhado gerenciado da configuração.

Recomendamos que você faça backup do armazenamento compartilhado regularmente para evitar a perda de dados. Para obter mais informações sobre como fazer backup de volumes individuais do Amazon EFS, consulte [Fazer backup dos sistemas de arquivos do Amazon EFS](#) no Guia do usuário do Amazon Elastic File System. Para saber como gerenciar backups de dados em Serviços da AWS, consulte [AWSBackup](#) no AWS Backup Guia do desenvolvedor.

Amazon FSx para Lustre

Para usar um sistema de arquivos FSx para Lustre externo existente para armazenamento permanente de longo prazo fora do ciclo de vida do cluster, especifique [FsxLustreSettings](#) / [FileSystemId](#).

Se você não especificar [FsxLustreSettings](#) / [FileSystemId](#), por padrão, AWS ParallelCluster cria um sistema de arquivos gerenciado do FSx para Lustre a partir de [FsxLustreSettings](#) no momento em que cria o cluster. AWS ParallelCluster também exclui o sistema de arquivos e os dados quando o cluster é excluído ou quando o sistema de arquivos é removido da configuração do cluster.

Para um sistema de arquivos do FSx para Lustre gerenciado pelo AWS ParallelCluster, você pode usar [FsxLustreSettings](#) / [DeletionPolicy](#) para instruir o AWS ParallelCluster a Delete ou Retain o sistema de arquivos quando o cluster for excluído ou quando o sistema de arquivos for removido da configuração do cluster. Por padrão, DeletionPolicy é definido como Delete.

Warning

Para armazenamento compartilhado gerenciado pelo AWS ParallelCluster, DeletionPolicy é definido como Delete por padrão.

Isso significa que, se uma das afirmações a seguir for verdadeira, um sistema de arquivos gerenciado e seus dados serão excluídos:

- Se você excluir o cluster.
- Se você alterar a configuração de armazenamento compartilhado gerenciado [SharedStorage](#) / [Name](#).
- Se você remover o armazenamento compartilhado gerenciado da configuração.

Recomendamos que você faça backup do armazenamento compartilhado regularmente para evitar a perda de dados. Você pode definir backups em seu cluster com [SharedStorage](#) / [FsxLustreSettings](#) / [AutomaticBackupRetentionDays](#) e [DailyAutomaticBackupStartTime](#). Para saber como gerenciar backups de dados em Serviços da AWS, consulte [AWSBackup](#) no AWS Backup Guia do desenvolvedor.

Configurar armazenamento compartilhado do FSx para ONTAP, FSx para OpenZFS e File Cache

Para FSx para ONTAP, FSx para OpenZFS e File Cache, você pode usar [FsxOntapSettings](#) / [VolumeId](#), [FsxOpenZfsSettings](#) / [VolumeId](#) e [FileCacheSettings](#) / [FileCacheId](#) para especificar a montagem de um volume externo existente ou cache de arquivos para seu cluster.

O armazenamento compartilhado gerenciado do AWS ParallelCluster não é compatível com FSx para ONTAP, FSx para OpenZFS e File Cache.

Trabalhando com armazenamento compartilhado no AWS ParallelCluster

Saiba mais sobre como trabalhar com o AWS ParallelCluster e com armazenamento compartilhado.

Tópicos

- [Considerações sobre armazenamento compartilhado do AWS ParallelCluster](#)
- [Converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo](#)

Considerações sobre armazenamento compartilhado do AWS ParallelCluster

Considere o seguinte ao trabalhar com armazenamento compartilhado no AWS ParallelCluster.

- Faça backup dos dados do sistema de arquivos com [AWS Backup](#) ou outro método para gerenciar backups em todos os seus sistemas de armazenamento.
- Para adicionar armazenamento compartilhado, adicione uma seção de armazenamento compartilhado ao seu arquivo de configuração e crie ou atualize o cluster.
- Para remover o armazenamento compartilhado, remova uma seção de armazenamento compartilhado do seu arquivo de configuração e atualize o cluster.
- Para substituir o armazenamento compartilhado gerenciado do AWS ParallelCluster existente pelo novo armazenamento gerenciado, altere o valor de [SharedStorage](#) / [Name](#) e atualize o cluster.

Warning

Por padrão, o armazenamento gerenciado do AWS ParallelCluster e os dados existentes são excluídos quando você executa a atualização do cluster com um novo parâmetro `Name`. Se você precisar alterar `Name` e reter os dados existentes do armazenamento compartilhado gerenciado, certifique-se de configurá-los `DeletionPolicy` como `Retain` ou faça backup dos dados antes de atualizar o cluster.

- Se você não fizer backup dos dados de armazenamento gerenciado do AWS ParallelCluster e `DeletionPolicy` for `Delete`, seus dados serão excluídos quando o cluster for excluído ou quando o armazenamento gerenciado for removido da configuração do cluster e o cluster for atualizado.
- Se você não fizer backup dos dados de armazenamento gerenciado do AWS ParallelCluster e `DeletionPolicy` for `Retain`, seu sistema de arquivos será desconectado antes do cluster ser excluído e poderá ser reconectado a outro cluster como um sistema de arquivos externo. Seus dados serão preservados.
- Se o armazenamento gerenciado do AWS ParallelCluster for removido da configuração do cluster e `DeletionPolicy` for `Retain`, ele poderá ser reconectado ao cluster como um sistema de arquivos externo com os dados do cluster preservados.

- A partir do AWS ParallelCluster versão 3.4.0, você pode aumentar a segurança das montagens do sistema de arquivos Amazon EFS definindo as configurações [SharedStorage](#) / [EfsSettings](#) / [EncryptionInTransit](#) e [IamAuthorization](#).
- Ao montar um sistema de arquivos externo no diretório /home, AWS ParallelCluster copia o conteúdo do diretório /home do nó principal para o sistema de arquivos externo. Ele copia dados existentes no diretório /home sem sobrescrever arquivos ou diretórios existentes no armazenamento externo. Isso inclui a chave SSH do cluster para o usuário padrão, caso ela ainda não exista no sistema de arquivos externo. Conseqüentemente, todos os outros clusters que montam o mesmo sistema de arquivos externo no respectivo diretório /home também terão a mesma chave SSH para o usuário padrão do cluster.
- Em um ambiente de vários clusters que monta o mesmo sistema de arquivos externo nos diretórios /home dos clusters, as chaves SSH que concedem acesso aos nós de computação, criadas no nó principal por AWS ParallelCluster, são geradas somente uma vez quando o primeiro cluster monta o sistema de arquivos externo em /home. Todos os outros clusters usam a mesma chave SSH. Como resultado, qualquer pessoa que possua a chave SSH do usuário padrão desses clusters compartilhados pode acessar qualquer cluster. Todos os nós de computação permitem conexões usando a chave gerada inicialmente.

Converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo

Saiba como converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo.

Os procedimentos são baseados no exemplo de trecho de arquivo de configuração a seguir.

```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Delete
...
```

Converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo

1. Defina `DeletionPolicy` para `Retain` no arquivo de configuração do cluster.


```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Retain
...
```

2. Para definir a alteração em `DeletionPolicy`, execute o seguinte comando.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

3. Remova a seção `SharedStorage` do arquivo de configuração do cluster.

```
...
...
```

4. Para alterar o `SharedStorage` gerenciado para `SharedStorage` externo e desconectá-lo do cluster, execute o comando a seguir.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. Seu armazenamento compartilhado agora é externo e separado do cluster.
6. Para conectar seu sistema de arquivos externo ao cluster original ou a outro cluster, siga estas etapas.
 - a. Obtenha o ID do sistema de arquivos do FSx para Lustre.

- i. Para usar o AWS CLI execute o comando a seguir e encontre o sistema de arquivos com um nome que inclua o nome do cluster original e anote o ID do sistema de arquivos.

```
aws fsx describe-file-systems
```

- ii. Para usar o AWS Management Console, faça login e navegue até <https://console.aws.amazon.com/fsx/>. Na lista de sistemas de arquivos, encontre o sistema de arquivos com um nome que inclua o nome do cluster original e anote o ID do sistema de arquivos.

- b. Atualize as regras do grupo de segurança do sistema de arquivos para fornecer acesso a partir e para o sistema de arquivos e as sub-redes do cluster. Você pode encontrar o nome e o ID do grupo de segurança do sistema de arquivos no console do Amazon FSx.

Adicione regras ao grupo de segurança do sistema de arquivos que permitam tráfego TCP de entrada e saída de e para o nó principal e os intervalos ou prefixos de CIDR IP do nó de computação. Especifique as portas TCP 988, 1021, 1022 e 1023 para o tráfego TCP de entrada e saída.

Para obter mais informações, consulte [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#) e [Criação, configuração e exclusão de grupos de segurança para o Amazon EC2](#) no Guia do usuário do AWS Command Line Interface versão 2.

- c. Adicione a seção SharedStorage à configuração do cluster.

```
...
- MountDir: /fsx
  Name: fsx-external
  StorageType: FsxLustre
  FsxLustreSettings:
    FileSystemId: fs-02e5b4b4abd62d51c
...
```

- d. Para adicionar o armazenamento compartilhado externo ao cluster, execute o comando a seguir.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

Cotas para armazenamento compartilhado

Configure o cluster SharedStorage para montar o armazenamento de arquivos compartilhado existente e criar um novo armazenamento de arquivos compartilhado com base nas cotas listadas na tabela a seguir.

As cotas de armazenamento de arquivos montados para cada cluster

Tipo de armazenamento compartilhado de arquivos	Armazenamento gerenciado do AWS ParallelCluster	Armazenamento externo	Total líquido da cota
Amazon EBS	5	5	5
RAID	1	0	1
Amazon EFS	1	20	21
Amazon FSx †	1 FSx para Lustre	20	21

Note

Essa tabela de cotas foi adicionada no AWS ParallelCluster versão 3.2.0.

† suporta AWS ParallelCluster somente a montagem dos sistemas Amazon FSx for NetApp ONTAP, Amazon FSx for OpenZFS e File Cache existentes. Ele não suporta a criação de novos sistemas FSx para ONTAP, FSx para OpenZFS e File Cache.

Note

Se você usar o AWS Batch como programador, o FSx para Lustre só ficará disponível no nó principal do cluster.

Os caches de arquivos não oferecem suporte a programadores AWS Batch.

Recursos do AWS ParallelCluster e marcação

Com o AWS ParallelCluster você pode criar tags para rastrear e gerenciar seus AWS ParallelCluster recursos. Você define as tags que deseja AWS CloudFormation criar e propagar para todos os recursos do cluster no [Seção Tags](#) do arquivo de configuração do cluster. Você também pode usar tags que o AWS ParallelCluster gera automaticamente para rastrear e gerenciar seus recursos.

Quando você cria um cluster, o cluster e seus recursos são marcados com as tags de sistema do AWS ParallelCluster e da AWS definidas nesta seção.

O AWS ParallelCluster aplica tags às instâncias, volumes e recursos do cluster. Para identificar a pilha do cluster, AWS CloudFormation aplica as tags de sistema da AWS às instâncias do cluster. Para identificar os modelos de execução do cluster EC2, o EC2 aplica tags de sistema às instâncias. Você pode usar tags para visualizar e gerenciar seus recursos no AWS ParallelCluster.

Não é possível modificar tags de sistema da AWS. Para evitar impactos na funcionalidade do AWS ParallelCluster, não modifique as tags do AWS ParallelCluster.

Veja a seguir um exemplo de uma tag do sistema AWS para um recurso do AWS ParallelCluster. Não é possível modificá-la.

```
"aws:cloudformation:stack-name"="clustername"
```

Veja a seguir um exemplo de uma tag do AWS ParallelCluster aplicada em um recurso. Não a modifique.

```
"parallelcluster:cluster-name"="clustername"
```

Você pode ver essas tags na seção EC2 do AWS Management Console.

Visualizar tags

1. Navegue até o console EC2 em <https://console.aws.amazon.com/ec2/>.
2. Para visualizar todas as tags do cluster, escolha Tags no painel de navegação.
3. Para visualizar as tags de cluster por instância, escolha Instâncias no painel de navegação.
4. Selecione uma instância de cluster.
5. Escolha a guia Gerenciar tags nos detalhes da instância e visualize as tags.
6. Escolha a guia Armazenamento nos detalhes da instância.
7. Selecione o ID do Volume.
8. Em Volumes, escolha o volume.
9. Escolha a guia Tags nos detalhes do volume e visualize as tags.

Tags de instância do nó principal do AWS ParallelCluster

Chave	Valor da tag
parallelcluster:cluster-name	<i>clustername</i>

Chave	Valor da tag
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>
aws:cloudformation:logical-id	HeadNode
aws:cloudformation:stack-id	arn:aws:cloudformation: <i>region-id</i> : <i>ACCOUNTID</i> :stack/ <i>clustername</i> / <i>1234abcd-12ab-12ab-12ab-1234567890abcdef0</i>
parallelcluster:version	<i>3.7.0</i>

Tags de volume raiz do nó principal do AWS ParallelCluster

Chave de tag	Valor da tag
parallelcluster:cluster-name	<i>clustername</i>
parallelcluster:node-type	HeadNode
parallelcluster:version	<i>3.7.0</i>

Tags de instância do nó de computação do AWS ParallelCluster

Chave	Valor da tag
parallelcluster:cluster-name	<i>clustername</i>
parallelcluster:compute-resource-name	<i>compute-resource-name</i>

Chave	Valor da tag
<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>3.7.0</i>

Tags de volume raiz do nó de computação do AWS ParallelCluster

Chave de tag	Valor da 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>3.7.0</i>

Tags de interface do usuário do AWS ParallelCluster

Chave de tag	Valor da tag
<code>parallelcluster-ui</code>	true

Monitoramento do AWS ParallelCluster e de logs

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do AWS ParallelCluster e das outras soluções da AWS. A AWS fornece as ferramentas

de monitoramento a seguir para observar o AWS ParallelCluster, informar quando algo está errado e realizar ações automaticamente quando apropriado:

- O Amazon CloudWatch monitora os recursos da AWS e as aplicações que você executa na AWS em tempo real. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode fazer o CloudWatch acompanhar o uso da CPU ou outras métricas das instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).
- O Amazon CloudWatch Logs permite monitorar, armazenar e acessar seus arquivos de log das instâncias do Amazon EC2, do CloudTrail e de outras fontes. O CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando determinados limites forem atingidos. Você também pode arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por sua conta da Conta da AWS ou em nome dela e entrega os arquivos de log a um bucket do Amazon S3 que você especifica. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).
- O Amazon EventBridge é um serviço de barramento de eventos sem servidor que facilita a conexão de aplicações a dados de diversas origens. O EventBridge fornece um fluxo de dados em tempo real de suas próprias aplicações, de aplicações de software como serviço (SaaS) e de serviços da AWS e roteia esses dados para destinos como o Lambda. Isso permite monitorar eventos que ocorrem em serviços e crie arquiteturas orientadas a eventos. Para obter mais informações, consulte o [Guia do usuário do Amazon EventBridge](#).

Tópicos

- [Integração com Amazon CloudWatch Logs](#)
- [Painel do Amazon CloudWatch](#)
- [Alarmes do Amazon CloudWatch para métricas de cluster](#)
- [Rotação de log configurado do AWS ParallelCluster](#)
- [Logs da CLI pcluster](#)
- [Logs de saída do console EC2](#)
- [Recupere registros de UI AWS ParallelCluster e tempo de execução AWS ParallelCluster](#)

- [Recuperando e preservando logs](#)

Integração com Amazon CloudWatch Logs

Para obter mais informações sobre o CloudWatch Logs, consulte o [Guia do usuário do Amazon CloudWatch Logs](#). Para configurar a integração com o CloudWatch Logs, consulte a seção [Monitoring](#). Para saber como acrescentar logs personalizados à configuração do CloudWatch usando `append-config`, consulte [Vários arquivos de configuração do atendente do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

Logs de cluster do Amazon CloudWatch Logs

Um grupo de logs é criado para cada cluster com um nome, `/aws/parallelcluster/cluster-name-<timestamp>` (por exemplo, `/aws/parallelcluster/testCluster-202202050215`). Cada log (ou conjunto de logs, se o caminho contiver um `*`) em cada nó possui um fluxo de logs denominado `{hostname}.{instance_id}.{logIdentifier}`. (Por exemplo, `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`). Os dados de log são enviados para o CloudWatch pelo [Atendente do CloudWatch](#), que é executado como `root` em todas as instâncias de cluster.

Um painel do Amazon CloudWatch é criado quando um cluster é criado. Esse painel permite revisar os logs armazenados no CloudWatch Logs. Para obter mais informações, consulte [Painel do Amazon CloudWatch](#).

Essa lista contém o `logIdentifier` e o caminho para os fluxos de log disponíveis para plataformas, programadores e nós.

Streams de log disponíveis para plataformas, programadores e nós

Plataformas	Programadores	Nós	Fluxos de log
amazon centos	awsbatc slurm	HeadNc	dcv-authenticator: <code>/var/log/parallelcluster/parallelcluster_dcv_authenticator.log</code>
redhat			dcv-ext-authenticator: <code>/var/log/parallelcluster/parallelcluster_dcv_connect.log</code>
ubuntu			dcv-agent: <code>/var/log/dcv/agent.*.log</code>

Plataformas	Programas	Nós	Fluxos de log
			dcv-xsession: /var/log/dcv/dcv-xsession.*.log dcv-server: /var/log/dcv/server.log dcv-session-launcher: /var/log/dcv/sessionlauncher.log Xdcv: /var/log/dcv/Xdcv.*.log cfn-init: /var/log/cfn-init.log chef-client: /var/log/chef-client.log
amazon centos redhat ubuntu	awsbatch slurm	Compute HeadNode	cloud-init: /var/log/cloud-init.log supervisord: /var/log/supervisord.log
amazon centos redhat ubuntu	slurm	Compute eet	cloud-init-output: /var/log/cloud-init-output.log computemgtd: /var/log/parallelcluster/computemgtd slurmd: /var/log/slurmd.log slurm_prolog_epilog: /var/log/parallelcluster/slurm_prolog_epilog.log

Plataformas	Programas	Nós	Fluxos de log
amazon centos redhat ubuntu	slurm	HeadNode	sssd: /var/log/sssds/sssds.log sssd_domain_default: /var/log/sssds/sssds_default.log pam_ssh_key_generator: /var/log/parallelcluster/pam_ssh_key_generator.log clusterstatusmgtd: /var/log/parallelcluster/clusterstatusmgtd clustermgtd: /var/log/parallelcluster/clustermgtd compute_console_output: /var/log/parallelcluster/compute_console_output slurm_resume: /var/log/parallelcluster/slurm_resume.log slurm_suspend: /var/log/parallelcluster/slurm_suspend.log slurmctld: /var/log/slurmctld.log slurm_fleet_status_manager: /var/log/parallelcluster/slurm_fleet_status_manager.log
amazon centos redhat	awsbatc slurm	Compute HeadNode	system-messages: /var/log/messages
ubuntu	awsbatc slurm	Compute HeadNode	syslog: /var/log/syslog

Trabalhos em clusters que usam AWS Batch armazenam a saída de trabalhos que atingiram um estado de RUNNING, SUCCEEDED ou FAILED no CloudWatch Logs. O grupo de logs é `/aws/batch/job`, e o formato do nome do fluxo de logs é `jobDefinitionName/default/ecs_task_id`. Por padrão, esses logs são configurados para nunca expirar, mas você pode modificar o período de retenção. Para obter mais informações, consulte [Alterar a retenção de dados de log no CloudWatch Logs](#) no Guia do usuário do Amazon CloudWatch Logs.

Logs de imagem de compilação do Amazon CloudWatch Logs

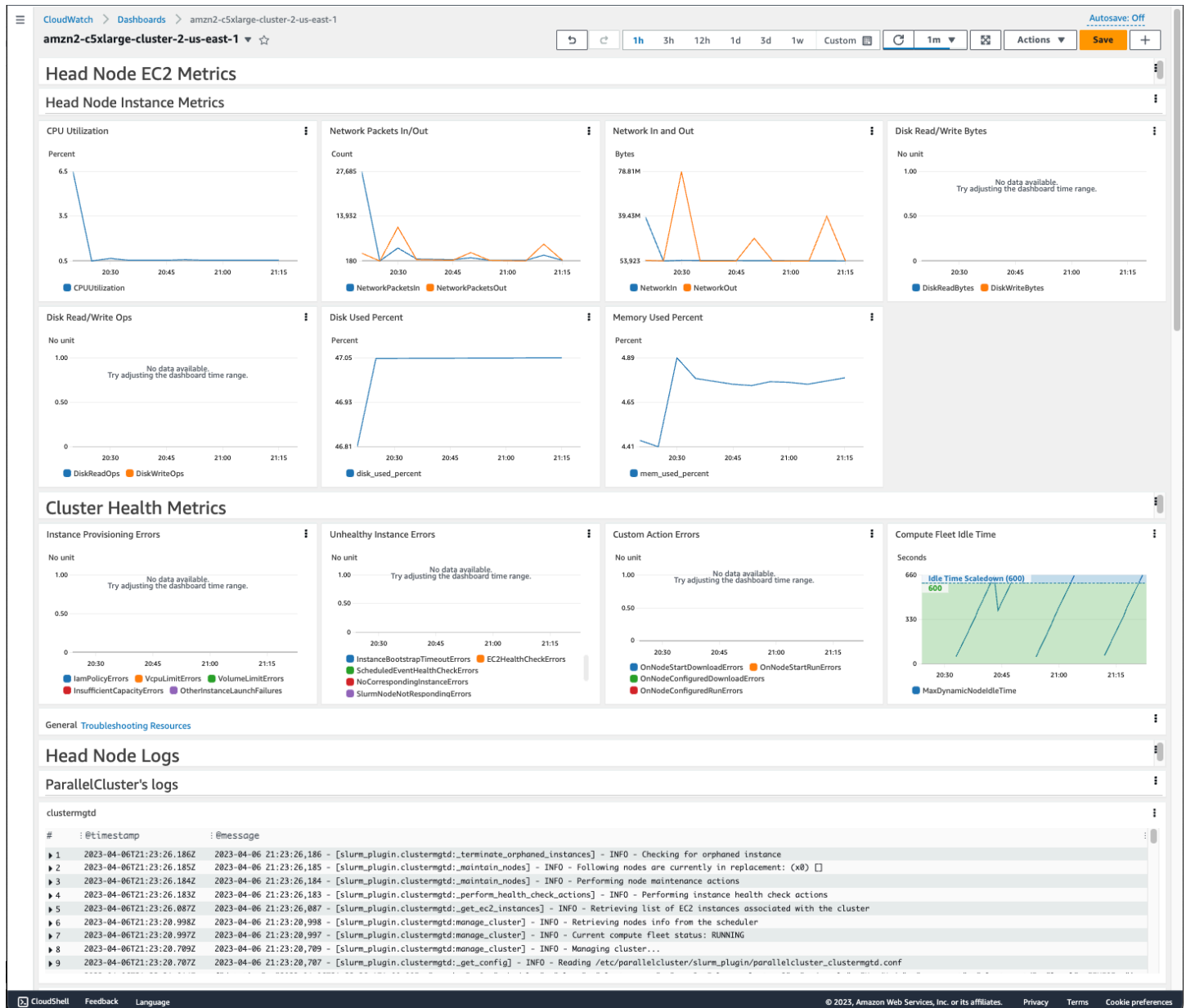
Um grupo de logs é criado para cada imagem de compilação personalizada com um nome, `/aws/imagebuilder/ParallelClusterImage-<image-id>`. Um fluxo de logs exclusivo com o nome `{pcluster-version}/1` contém a saída do processo da imagem de compilação.

Você pode acessar os logs usando os comandos de imagem do `pcluster`. Para obter mais informações, consulte [AWS ParallelCluster Personalização da AMI](#).

Painel do Amazon CloudWatch

Um painel do Amazon CloudWatch é criado quando um cluster é criado. Isso facilita o monitoramento dos nós em seu cluster e a visualização dos logs armazenados no Amazon CloudWatch Logs. O nome do painel é `ClusterName-Region`. `ClusterName` é o nome do seu cluster e `Region` é a Região da AWS em que o cluster está. Você pode acessar o painel no console ou abrindo `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`.

A imagem a seguir mostra um exemplo de painel do CloudWatch para cluster.



Métricas de instância do nó principal

A primeira seção do painel exibe gráficos das métricas do EC2 do nó principal.

Se seu cluster tiver armazenamento compartilhado, a próxima seção mostrará métricas de armazenamento compartilhado.

Métricas de integridade do cluster

Se seu cluster usa Slurm para programação, os gráficos de métrica de integridade do cluster mostram erros em tempo real do nó de computação do cluster. Para obter mais informações,

consulte [Métricas de integridade do cluster para solução de problemas](#). As métricas de integridade do cluster foram adicionadas ao painel a partir do AWS ParallelCluster versão 3.6.0.

Logs do nó principal

A seção final lista os logs do nó principal agrupados por logs do AWS ParallelCluster, logs do programador, logs de integração do NICE DCV e logs do sistema.

Para obter mais informações sobre painéis do Amazon CloudWatch, consulte [Como usar painéis do Amazon CloudWatch](#), no Guia do usuário do Amazon CloudWatch.

Se você não quiser criar o painel do Amazon CloudWatch, você pode desativá-lo definindo [Monitoring](#) / [Dashboards](#) / [CloudWatch](#) / [Enabled](#) para `false`.

Note

Se você desativar a criação do painel do Amazon CloudWatch, você também desativa `disk_used_percent` e `memory_used_percent` do Amazon CloudWatch e os alarmes do seu cluster. Para obter mais informações, consulte [Alarmes do Amazon CloudWatch para métricas de cluster](#).

Os alarmes `disk_used_percent` e `memory_used_percent` são adicionados a partir do AWS ParallelCluster versão 3.6.

Alarmes do Amazon CloudWatch para métricas de cluster

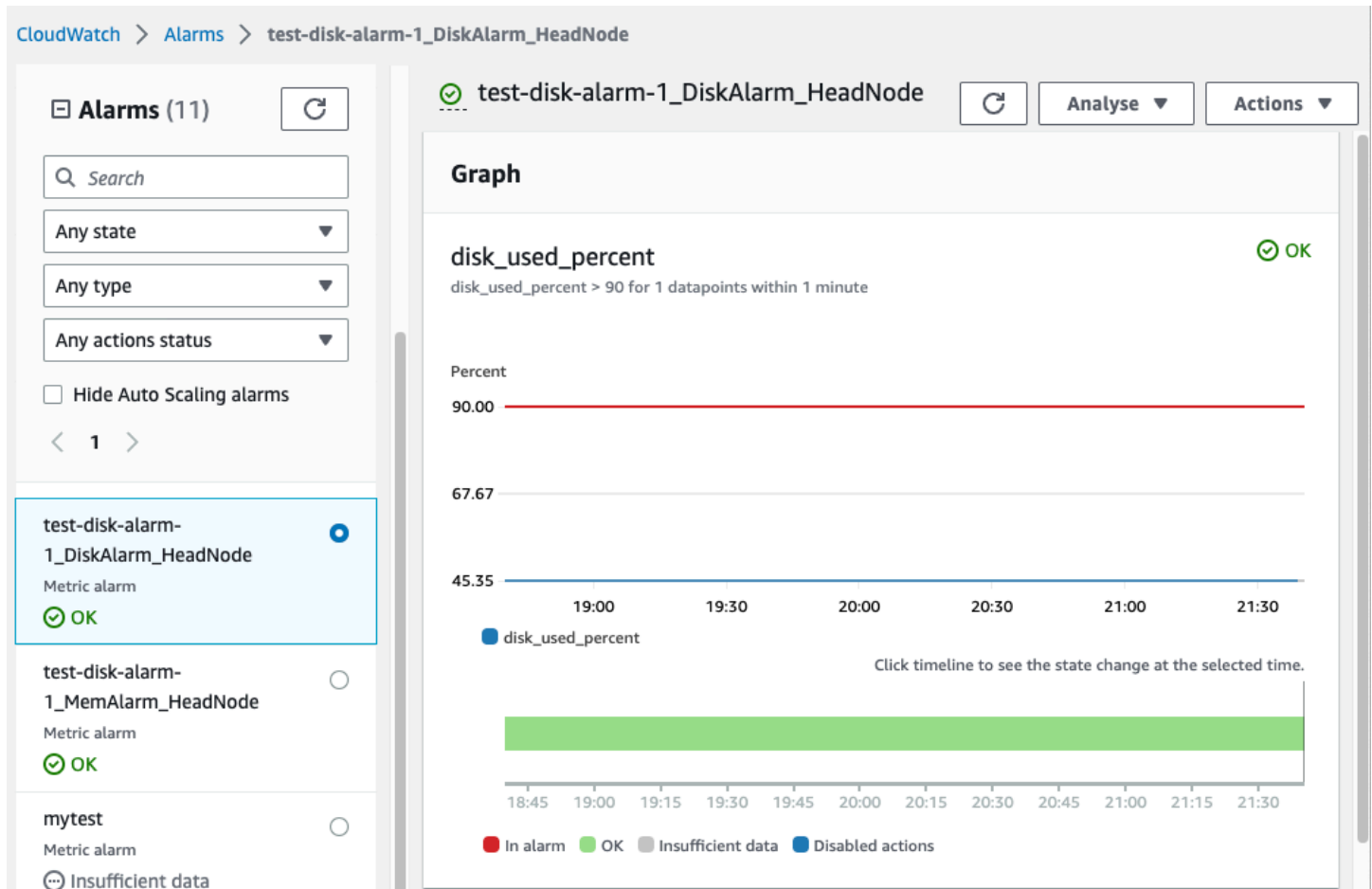
A partir do AWS ParallelCluster versão 3.6, você pode configurar seu cluster com alarmes do Amazon CloudWatch para monitorar o nó principal. Um alarme monitora o volume raiz `disk_used_percent`. O outro alarme monitora a métrica `mem_used_percent`. Para obter mais informações, consulte [Métricas coletadas pelo atendente do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

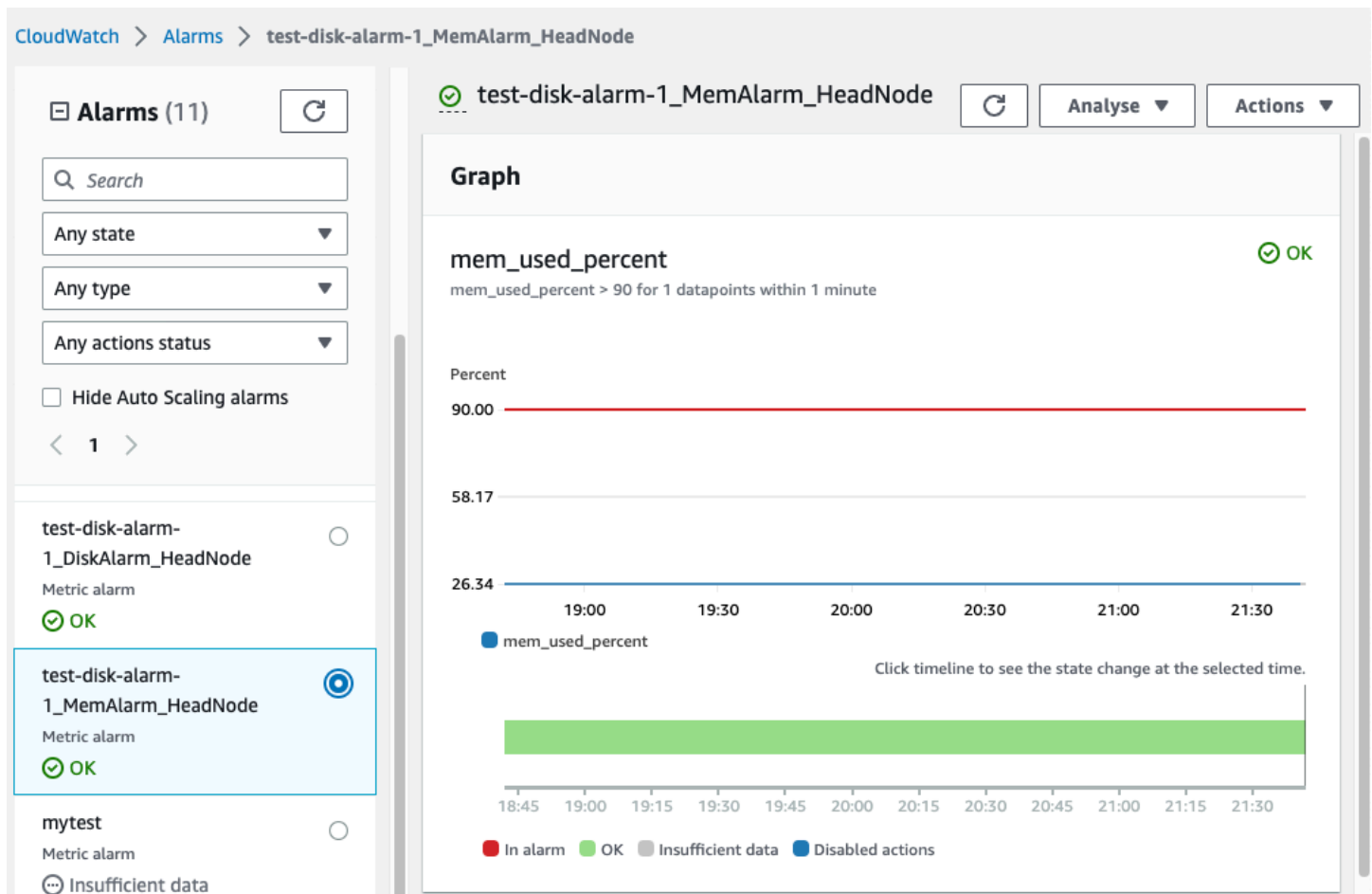
Os alarmes são nomeados da seguinte forma:

- `cluster-name_DiskAlarm_HeadNode`
- `cluster-name_MemAlarm_HeadNode`

`cluster-name` é o nome do seu cluster.

Para acessar os alarmes no console do CloudWatch escolha Alarmes no painel de navegação. As imagens a seguir mostram o alarme de uso do disco e o alarme de uso da memória de um cluster.





O alarme de uso do disco está no estado ALARM quando a porcentagem de uso do disco é maior que 90% para 1 ponto de dados, dentro do período de 1 minuto.

O alarme de uso da memória está no estado ALARM quando a porcentagem de uso da memória é maior que 90% para 1 ponto de dados, dentro do período de 1 minuto.

Note

O AWS ParallelCluster não configura ações de alarme por padrão. Para obter informações sobre como configurar ações de alarme, como enviar notificações, consulte [Ações de alarme](#). Para obter mais informações sobre alarmes do Amazon CloudWatch, consulte [Como usar alarmes do Amazon CloudWatch](#), no Guia do usuário do Amazon CloudWatch.

Se você não quiser criar esses alarmes do Amazon CloudWatch, desative-os definindo [Monitoring / Dashboards / CloudWatch / Enabled](#) para `false` na configuração do cluster. Isso também

desativa a criação do painel do Amazon CloudWatch. Para obter mais informações, consulte [Painel do Amazon CloudWatch](#).

Note

Se você desativar a criação do painel do Amazon CloudWatch, você também desativa `disk_used_percent` e `memory_used_percent` do Amazon CloudWatch e os alarmes do seu cluster.

Rotação de log configurado do AWS ParallelCluster

As configurações de rotação de log do AWS ParallelCluster estão localizadas nos arquivos `/etc/logrotate.d/parallelcluster*_log_rotation`. Quando um log configurado é rotacionado, o conteúdo do log atual é preservado em um único backup e o log vazio retoma o registro.

Somente 1 backup é mantido para cada log configurado.

O AWS ParallelCluster configura um log de rápido crescimento para alternar quando ele atinge 50 MB de tamanho. Os log de rápido crescimento estão relacionados ao dimensionamento e ao Slurm, incluindo `/var/log/parallelcluster/clustermgtd`, `/var/log/parallelcluster/slurm_resume.log` e `/var/log/slurmctld.log`.

O AWS ParallelCluster configura um log de crescimento lento para alternar quando ele atinge 10 MB de tamanho.

Você pode visualizar logs anteriores que são retidos pelo número de dias definido na configuração do cluster [Logs](#) / [CloudWatch](#) / [RetentionInDays](#) com o registro em log do CloudFormation ativado. Verifique as configurações `RetentionInDays` para ver se o número de dias precisa ser aumentado para seu caso de uso.

O AWS ParallelCluster configura e alterna os seguintes logs:

Logs do nó principal

```
/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
```

Logs de nós de computação

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log
```

Logs do nó de login

```
/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
```

Logs da CLI **pcluster**

A CLI **pcluster** grava logs de seus comandos em arquivos `pcluster.log.#` em `/home/user/.parallelcluster/`.

Para cada comando, os logs geralmente incluem o comando com entradas, uma cópia da versão da API CLI usada para criar o comando, a resposta e as informações e mensagens de erro. Para um comando de criação e construção, os logs também incluem o arquivo de configuração, as operações de validação do arquivo de configuração, o modelo do CloudFormation e os comandos de pilha.

Você pode usar esses logs para verificar erros, entradas, versões e comandos da CLI **pcluster**. Eles também podem servir como um registro de quando os comandos foram feitos.

Logs de saída do console EC2

Quando AWS ParallelCluster detecta que uma instância de nó de computação estática é encerrada inesperadamente, ela tenta recuperar a saída do console EC2 da instância do nó encerrado após um período de tempo. Dessa forma, se o nó de computação não conseguir se comunicar com o Amazon CloudWatch, informações úteis de solução de problemas sobre o motivo pelo qual o nó foi encerrado ainda poderão ser recuperadas da saída do console. Essa saída do console é gravada no log `/var/log/parallelcluster/compute_console_output` no nó principal. Para obter mais informações sobre a saída do console do EC2, consulte [Saída do console da instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Por padrão, o AWS ParallelCluster só recupera a saída do console de um subconjunto de amostra de nós encerrados. Isso evita que o nó principal do cluster fique sobrecarregado com várias solicitações de saída do console causadas por um grande número de encerramentos. Por padrão, o AWS ParallelCluster aguarda 5 minutos entre a detecção de encerramento e a recuperação da saída do console para dar tempo ao EC2 para recuperar a saída final do console dos nós.

Você pode editar o tamanho da amostra e os valores dos parâmetros de tempo de espera no arquivo `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` no nó principal.

Esse atributo foi adicionado no AWS ParallelCluster versão 3.5.0.

Parâmetros de saída do console do EC2

Você pode editar os valores dos seguintes parâmetros de saída do console EC2 no arquivo `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` no nó principal.

compute_console_logging_enabled

Para desativar a coleta de log de saída do console, defina `compute_console_logging_enabled` como `false`. O padrão é `true`.

Você pode atualizar esse parâmetro a qualquer momento, sem interromper a frota de computação.

compute_console_logging_max_sample_size

`compute_console_logging_max_sample_size` define o número máximo de nós de computação dos quais o AWS ParallelCluster coleta as saídas do console sempre que detecta um encerramento inesperado. Se esse valor for menor que 1, o AWS ParallelCluster recupera a saída do console de todos os nós encerrados. O valor padrão é 1.

Você pode atualizar esse parâmetro a qualquer momento, sem interromper a frota de computação.

compute_console_wait_time

`compute_console_wait_time` define o tempo de espera do AWS ParallelCluster, em segundos, entre a detecção de uma falha no nó e a coleta da saída do console desse nó. Você pode aumentar o tempo de espera se determinar que o EC2 precisa de mais tempo para coletar a saída final do nó encerrado. O valor padrão é de 300 segundos (5 minutos).

Você pode atualizar esse parâmetro a qualquer momento, sem interromper a frota de computação.

Recupere registros de UI AWS ParallelCluster e tempo de execução AWS ParallelCluster

Saiba como recuperar os registros da interface do usuário AWS ParallelCluster e do tempo de execução AWS ParallelCluster para solucionar problemas. Para começar, encontre os nomes relevantes da interface AWS ParallelCluster e da pilha AWS ParallelCluster. Use o nome da pilha para localizar os grupos de registros de instalação. Para finalizar, exporte os registros. Esses registros são específicos para o tempo de execução AWS ParallelCluster. Para clusters de logs, consulte [Recuperando e preservando logs](#).

Pré-requisitos

- O AWS CLI está instalado.
- Você tem credenciais para executar comandos AWS CLI no Conta da AWS qual a interface do usuário AWS ParallelCluster está ativada.
- Você pode acessar o console do Amazon CloudWatch no local em que Conta da AWS a interface do usuário está AWS ParallelCluster ativada.

Etapa 1: Localize os nomes das pilhas relevantes

No exemplo a seguir, substitua o texto destacado em vermelho com seus valores reais.

Liste as pilhas, usando o local Região da AWS em que você instalou a interface do usuário AWS ParallelCluster:

```
$ aws cloudformation list-stacks --region aws-region-id
```

Observe os nomes das pilhas a seguir:

- O nome da pilha que implantou a interface do usuário AWS ParallelCluster na sua conta. Você inseriu esse nome ao instalar a interface do usuário AWS ParallelCluster; por exemplo, `pcluster-ui`.
- A pilha AWS ParallelCluster que é prefixada com o nome da pilha que você inseriu; por exemplo, `pcluster-ui-ParallelClusterApi-ABCD1234EFGH`

Etapa 2: Localizar os grupos de logs.

Liste os grupos de log da pilha de UI AWS ParallelCluster, conforme mostrado no exemplo a seguir:

```
$ 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
```

Liste os grupos de log da pilha de API AWS ParallelCluster, conforme mostrado no exemplo a seguir:

```
$ aws cloudformation describe-stack-resources \
  --region aws-region-id \
  --stack-name pcluster-ui-ParallelClusterApi-ABCD1234EFGH \
  --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' && LogicalResourceId
== 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \
  --output text
```

Anote as listas de grupos de logs para uso na próxima etapa.

Etapa 3: exportar os logs

Siga as etapas a seguir para coletar e exportar os logs:

1. Faça login no AWS Management Console, em seguida, navegue até o console do [Amazon CloudWatch](#) no Conta da AWS no qual a interface do usuário AWS ParallelCluster está ativada.
2. Escolha Logs e selecione Logs Insights.
3. Selecione todos os grupos de logs listados na etapa anterior.

- Escolha um intervalo de tempo, como 12 horas.
- Execute a seguinte consulta:

```
$ fields @timestamp, @message  
| sort @timestamp desc  
| limit 10000
```

- Escolha Export results (Exportar resultados), Download table (Baixar tabela) (JSON).

Recuperando e preservando logs

O AWS ParallelCluster cria métricas do EC2 para armazenamento e instâncias de computação e HeadNode. É possível visualizar as métricas nos Painéis personalizados do console do CloudWatch. O AWS ParallelCluster também cria fluxo de logs do CloudWatch em cluster nos grupos de logs. Você pode visualizar esses logs nos Painéis personalizados do console CloudWatch ou em Grupos de logs. A seção [Monitoramento](#) da configuração do cluster descreve como você pode modificar os logs e o painel do CloudWatch do cluster. Para obter mais informações, consulte [Integração com Amazon CloudWatch Logs](#) e [Painel do Amazon CloudWatch](#).

Os logs são um recurso útil para solucionar problemas. Por exemplo, se você quiser excluir um cluster com falha, talvez seja útil criar primeiro um arquivo dos logs do cluster. Siga as etapas [Arquivar logs](#) para criar um arquivo.

Tópicos

- [Logs de cluster indisponíveis no CloudWatch](#)
- [Arquivar logs](#)
- [Logs preservados](#)
- [Logs de nós encerrados](#)

Logs de cluster indisponíveis no CloudWatch

Se os logs do cluster não estiverem disponíveis no CloudWatch, verifique se você não substituiu a configuração do AWS ParallelCluster de log do CloudWatch ao adicionar logs personalizados à configuração.

Para adicionar logs personalizados à configuração do CloudWatch, certifique-se de anexar à configuração em vez de buscá-la e substituí-la. Para obter mais informações sobre `fetch-config`

e `append-config`, consulte [Vários arquivos de configuração do atendente do CloudWatch](#) no Guia do usuário do CloudWatch.

Para restaurar a configuração do AWS ParallelCluster de log do CloudWatch, você pode executar os seguintes comandos dentro de um nó do AWS ParallelCluster:

```
$ 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
```

Arquivar logs

Você pode arquivar os logs no Amazon S3 ou em um arquivo local (dependendo do parâmetro `--output-file`).

Note

Adicione permissões à política de bucket do Amazon S3 para conceder acesso ao CloudWatch. Para obter mais informações, consulte [Definir permissões em um bucket do Amazon S3](#) no Guia do usuário do Amazon CloudWatch Logs.

```
$ 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"
```

```
}
```

O arquivo contém os fluxos do Amazon CloudWatch Logs e eventos de pilha do AWS CloudFormation oriundos do nó principal e dos nós de computação dos últimos 14 dias, a menos que especificado explicitamente na configuração ou nos parâmetros do comando `export-cluster-logs`. O tempo necessário para a conclusão do comando depende do número de nós no cluster e do número de fluxos de log disponíveis no CloudWatch Logs. Para obter mais informações sobre os fluxos de log disponíveis, consulte [Integração com Amazon CloudWatch Logs](#).

Logs preservados

A partir da versão 3.0.0, o AWS ParallelCluster preserva os logs do CloudWatch por padrão quando um cluster é excluído. Se você quiser excluir um cluster e preservar seus logs, certifique-se de que [Monitoring / Logs / CloudWatch / DeletionPolicy](#) não esteja definido como Delete na configuração do cluster. Caso contrário, altere o valor desse campo para Retain e execute o comando `pcluster update-cluster`. Em seguida, execute `pcluster delete-cluster --cluster-name <cluster_name>` para excluir o cluster, mas retenha o grupo de logs que está armazenado no Amazon CloudWatch.

Logs de nós encerrados

Se um nó de computação estático for encerrado inesperadamente e o CloudWatch não tiver logs, verifique se o AWS ParallelCluster gravou a saída do console desse nó de computação no nó principal do log `/var/log/parallelcluster/compute_console_output`. Para obter mais informações, consulte [Logs principais para depuração](#).

Se o log `/var/log/parallelcluster/compute_console_output` não estiver disponível ou não contiver a saída do nó, use o AWS CLI para recuperar a saída do console a partir do nó com falha. Faça login no nó principal do cluster e obtenha o nó com falha `instance-id` do arquivo `/var/log/parallelcluster/slurm_resume.log`.

Recupere a saída do console usando o comando a seguir com o `instance-id`:

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Se um nó de computação dinâmico for encerrado automaticamente após o lançamento e o CloudWatch não tiver logs em relação a isso, envie um trabalho que ative uma ação de dimensionamento de cluster. Aguarde até que a instância falhe e recupere o log do console da instância.

Faça login no nó principal do cluster e obtenha o nó de computação `instance-id` do arquivo `/var/log/parallelcluster/slurm_resume.log`.

Recupere o log do console da instância usando o comando a seguir:

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

O log de saída do console pode ajudá-lo a depurar a causa raiz de uma falha no nó de computação quando o log do nó de computação não estiver disponível.

AWS CloudFormation recurso personalizado

A partir da AWS ParallelCluster versão 3.6.0, você pode usar um recurso AWS ParallelCluster CloudFormation personalizado em uma AWS CloudFormation pilha. O recurso personalizado é uma pilha AWS ParallelCluster hospedada. Dessa forma, você pode usar CloudFormation para configurar e gerenciar seus clusters. Por exemplo, você pode configurar recursos externos do cluster, como rede, armazenamento compartilhado e infraestrutura de grupo de segurança em uma CloudFormation pilha. Além disso, você pode gerenciar seu cluster com uma CloudFormation infraestrutura como pipeline de código.

Adicione um recurso AWS ParallelCluster personalizado ao seu CloudFormation modelo fazendo o seguinte:

1. Adicione uma pilha personalizada de provedores de recursos que seja de propriedade e hospedada por AWS ParallelCluster.
2. Faça referência à pilha de provedores em seu CloudFormation modelo como um recurso personalizado.

A pilha de provedores de recursos personalizados trata e responde às CloudFormation solicitações. Por exemplo, ao implantar sua CloudFormation pilha, você também configura e cria um cluster. Para atualizar um cluster, você atualiza sua CloudFormation pilha. Você exclui um cluster ao excluir sua pilha. Para obter mais informações sobre recursos CloudFormation personalizados, consulte [Recursos personalizados](#) no Guia do AWS CloudFormation usuário.

Warning

CloudFormation não detecta desvios de recursos personalizados. Use somente CloudFormation para atualizar a configuração do cluster e excluir um cluster.

Você pode usar a CLI [pcluster](#) ou [Interface do usuário AWS ParallelCluster](#) para monitorar o estado do cluster ou para atualizar a frota de computação, mas não deve usá-las para atualizar a configuração do cluster ou excluir o cluster.

Note

Recomendamos que você adicione [proteção contra encerramento](#) à sua pilha para evitar a remoção acidental.

Pilha de provedores hospedada por AWS ParallelCluster

A pilha de provedores de recursos personalizados é formatada conforme mostrado no seguinte trecho de CloudFormation modelo:

```
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: 3.7.0 }
```

Propriedades:

Parâmetros:

CustomLambdaRole (opcional):

Um perfil personalizado com permissões para executar o AWS Lambda que cria e gerencia o cluster. Por padrão, o perfil usa as mesmas políticas definidas por padrão na [documentação do AWS ParallelCluster](#).

AdditionalIamPolicies (opcional):

Uma lista separada por vírgulas de nomes de recursos da Amazon (ARNs) adicionais da política do IAM para adicionar ao perfil usado pelo Lambda. Isso só é usado se não for especificado um CustomLambdaRole e se puder ser mantido em branco.

Se você precisar de políticas adicionais para o nó principal, nós de computação ou para acessar um bucket do Amazon S3, adicione-as às propriedades `CustomLambdaRole` ou `AdditionalIamPolicy`.

Para obter mais informações sobre políticas gerenciadas, consulte [AWS Identity and Access Management permissões em AWS ParallelCluster](#).

TemplateURL (obrigatório):

O URL do arquivo de recurso AWS ParallelCluster personalizado.

Saídas:

ServiceToken:

Um valor que pode ser usado como uma propriedade de recurso personalizada `ServiceToken`. Um recurso personalizado `ServiceToken` especifica para onde AWS CloudFormation envia as solicitações. Essa é uma entrada obrigatória para um recurso de cluster que você inclui em seu AWS CloudFormation modelo.

LogGroupArn:

O ARN do CloudWatch LogGroup qual o recurso subjacente se registra.

LambdaLayerArn:

O ARN da camada Lambda usada para executar operações. AWS ParallelCluster

Recurso do cluster

O recurso de CloudFormation cluster é formatado conforme mostrado no seguinte trecho CloudFormation de modelo:

```
PclusterCluster:
  Type: Custom::PclusterCluster
  Properties:
    ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
    ClusterName: !Sub 'c-${AWS::StackName}' # Must be different from StackName
    ClusterConfiguration:
      # Your Cluster Configuration
```

Propriedades:

ServiceToken:

A `ServiceToken` saída da pilha do AWS ParallelCluster provedor.

ClusterName:

O nome do cluster a ser criado e gerenciado. O nome não deve corresponder ao nome da CloudFormation pilha. O nome não pode ser alterado após a criação do cluster.

ClusterConfiguration:

O arquivo YAML de configuração do cluster, conforme descrito em [Arquivo de configuração do cluster](#). No entanto, você pode usar as CloudFormation construções usuais, como funções [intrínsecas](#).

DeletionPolicy:

Define se o cluster deve ser excluído quando a pilha raiz for excluída. O padrão é `Delete`.

Retain:

Mantém o cluster se o recurso personalizado for excluído.

Note

Para manter o cluster retido funcionando, os recursos dependentes do cluster, como armazenamento e rede, devem ter uma política de exclusão definida para reter.

Delete:

Exclui o cluster se o recurso personalizado for excluído.

Valores de retorno **Fn::GetAtt**:

A função intrínseca `Fn::GetAtt` retorna um valor para um atributo especificado de um tipo. Para obter mais informações sobre o uso da `Fn::GetAtt` intrínseca função, consulte [Fn::GetAtt](#).

ClusterProperties:

Os valores da operação [pcluster describe-cluster](#).

validationMessages:

Uma string contendo todas as mensagens de validação que ocorreram durante a última operação de criação ou atualização.

logGroupName:

O nome do grupo de logs usado para registrar as operações do cluster Lambda. Os eventos de log são retidos por 90 dias e o grupo de log é retido após a exclusão do cluster.

Exemplo: 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 ]
```

Exemplo: CloudFormation modelo simples e completo com um recurso AWS ParallelCluster personalizado:

```
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: 3.7.0 }
```

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

Para saber mais sobre como usar o recurso CloudFormation AWS ParallelCluster personalizado, consulte [Criação de um cluster com AWS CloudFormation](#).

Operações de cluster

Quando um recurso personalizado de cluster é adicionado a uma CloudFormation pilha, CloudFormation pode realizar as seguintes operações de cluster:

- CloudFormation cria um cluster em uma nova pilha separada ao implantar uma pilha que inclui o AWS ParallelCluster recurso personalizado.
- Se você atualizar a configuração do cluster definida na pilha, de acordo com as políticas de atualização de configuração, CloudFormation atualizará o cluster. O provedor de recursos AWS ParallelCluster personalizados não interrompe a frota de computação antes de atualizar o cluster. Recomendamos usar a configuração [QueueUpdateStrategy](#) para atualizações do cluster. Dessa forma, você pode evitar fazer `pcluster update-compute-fleet` chamadas explícitas antes e depois das atualizações ao usar o recurso AWS ParallelCluster personalizado.
- Se você excluir a pilha, o cluster será excluído.

Pilhas de solução de problemas que incluem o recurso AWS ParallelCluster personalizado

Com um recurso AWS ParallelCluster personalizado, CloudFormation implanta um cluster a partir de uma pilha nova e separada. Você pode monitorar a criação do cluster seguindo estas etapas:

1. Navegue até CloudFormation AWS Management Console e escolha Pilhas no painel de navegação.
2. Escolha a pilha com o nome que você definiu para o nome do cluster.
3. Se o estado da pilha for `ROLLBACK_COMPLETE`, ocorreu um erro durante a criação do cluster.
4. Escolha Detalhes da pilha e escolha a guia Eventos.
5. Pesquise Eventos na ID lógica pelo nome que você definiu para o nome do cluster. Ela contém um `Status reason` que especifica uma razão para um problema.
6. Você também pode escolher o menu suspenso Pilhas e, em seguida, Excluídas, para ver a lista de pilhas excluídas. Selecione a pilha com o nome do cluster e visualize Eventos para obter mais detalhes.
7. Para visualizar a saída do provedor de recursos personalizados que gerencia o cluster, selecione a pilha com a descrição "Recurso personalizado do AWS ParallelCluster cluster". Escolha a guia Recursos, encontre o recurso com ID lógica `PclusterCfnFunctionLogGroup` e acesse o link fornecido. Veja os fluxos de log que mostram a saída de depuração do Lambda.

8. Para solucionar problemas do cluster, consulte [Solução de problemas do AWS ParallelCluster](#).

Elastic Fabric Adapter

O Elastic Fabric Adapter (EFA) é um dispositivo de rede que tem recursos de OS-bypass para comunicações de rede de baixa latência com outras instâncias na mesma sub-rede. O EFA é exposto usando Libfabric e pode ser usado por aplicativos que usam o padrão MPI (Messaging Passing Interface).

Para usar o EFA com AWS ParallelCluster e um programador Slurm, defina [SlurmQueues](#) / [ComputeResources](#) / [Efa](#) / [Enabled](#) como true.

Para visualizar uma lista de tipos de instâncias do EC2 compatíveis com EFAs, consulte [Tipos de instância compatíveis](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Recomendamos que você execute suas instâncias habilitadas para EFA em um grupo de posicionamento. Dessa forma, as instâncias são executadas em um grupo de baixa latência em uma única zona de disponibilidade. Para obter mais informações sobre como configurar grupos de posicionamento com AWS ParallelCluster, consulte [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#).

Para obter mais informações, consulte [Elastic Fabric Adapter](#) no Guia do usuário do Amazon EC2 para instâncias do Linux e [Dimensionar cargas de trabalho HPC com o Elastic Fabric Adapter e o AWS ParallelCluster](#) no Blog sobre código aberto da AWS.

Note

O Elastic Fabric Adapter (EFA) não é compatível com diferentes zonas de disponibilidade. Para obter mais informações, consulte [Scheduling](#) / [SlurmQueues](#) / [Networking](#) / [SubnetIds](#).

Note

Por padrão, as distribuições Ubuntu habilitam a proteção ptrace (rastreamento do processo). A proteção ptrace fica desativada para que o Libfabric funcione corretamente. Para obter mais informações, consulte [Desativação da proteção ptrace](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Habilitar o Intel MPI

O Intel MPI está disponível nas AMIs do AWS ParallelCluster para os valores de `alinux2`, `centos7`, `rhel8`, `ubuntu2204` e `ubuntu2004` para a configuração [Image](#) / [Os](#).

Note

Para usar o Intel MPI, você deve concordar e aceitar os termos da [Licença de Software Simplificada da Intel](#).

Por padrão, o Open MPI é colocado no caminho. Para habilitar o Intel MPI em vez do Open MPI, o módulo Intel MPI deve ser carregado primeiro. Em seguida, você deve instalar a versão mais recente usando `module load intelmpi`. O nome exato do módulo muda com cada atualização. Para ver quais módulos estão disponíveis, execute `module avail`. A saída é a seguinte:

```
$ 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
```

Para carregar um módulo, execute `module load modulename`. Você pode adicionar isso ao script usado para executar `mpirun`.

```
$ module load intelmpi
```

Para ver quais módulos estão carregados, execute `module list`.

```
$ module list
Currently Loaded Modulefiles:
 1) intelmpi
```

Para verificar se o Intel MPI está habilitado, execute `mpirun --version`.


```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2021.6 Build 20220227 (id: 28877f3f32)
Copyright 2003-2022, Intel Corporation.
```

Depois que o módulo Intel MPI é carregado, vários caminhos são alterados para usar as ferramentas Intel MPI. Para executar o código compilado pelas ferramentas Intel MPI, carregue o módulo Intel MPI primeiro.

Note

O Intel MPI não é compatível com instâncias baseadas em Graviton da AWS.

Note

Antes do AWS ParallelCluster versão 2.5.0, o Intel MPI não estava disponível nas AMIs do AWS ParallelCluster nas regiões China (Pequim) e China (Ningxia).

API AWS ParallelCluster

O que é a API AWS ParallelCluster?

A API AWS ParallelCluster é um aplicativo sem servidor que, uma vez implantado em sua Conta da AWS, fornece acesso programático aos atributos do AWS ParallelCluster por meio de uma API.

A API AWS ParallelCluster é distribuída como um modelo [AWS CloudFormation](#) independente, que inclui um endpoint do [Amazon API Gateway](#), que expõe atributos do AWS ParallelCluster, e uma função [AWS Lambda](#), que cuida do processamento dos atributos invocados.

A imagem a seguir mostra um diagrama da arquitetura de alto nível da infraestrutura da API AWS ParallelCluster.

Documentação da API AWS ParallelCluster

O arquivo de especificação da OpenAPI que descreve a API AWS ParallelCluster pode ser baixado em:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

A partir do arquivo de especificação da OpenAPI, você pode gerar documentação para a API AWS ParallelCluster usando uma das muitas ferramentas disponíveis, como o [Swagger UI](#) ou [Redoc](#).

Como implantar uma API AWS ParallelCluster

Para implantar a API AWS ParallelCluster, você precisa ser Administrador da Conta da AWS.

O modelo usado para implantar a API está disponível no seguinte URL:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

onde **<REGION>** é a Região da AWS onde a API precisa ser implantada e **<VERSION>** é a versão da AWS ParallelCluster (por exemplo, 3.7.0).

AWS Lambda processa os atributos invocados pela API usando uma interface de camada Lambda com a [API da biblioteca Python do AWS ParallelCluster](#).

Warning

Qualquer usuário na Conta da AWS, que tenha acesso privilegiado aos serviços AWS Lambda do Amazon API Gateway, herda automaticamente as permissões para administrar os atributos da API AWS ParallelCluster.

Implantação com AWS CLI

Configure as credenciais da AWS para serem usadas com a CLI, caso ainda não tenha configurado.

```
$ aws configure
```

Execute o comando a seguir para implantar a API.

```
$ REGION=<region>  
$ API_STACK_NAME=<stack-name> # This can be any name  
$ VERSION=3.7.0  
$ 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}
```

Personalize sua implantação

Você pode personalizar a implantação da API usando os parâmetros da AWS CloudFormation expostos pelo modelo. Para configurar o valor de um parâmetro ao implantar por meio da CLI, a seguinte opção pode ser usada: `--parameters ParameterKey=KeyName,ParameterValue=Value`.

Os seguintes parâmetros são opcionais:

- **Região** - Use o parâmetro `Region` para especificar se a API é capaz de controlar os recursos em todas as Regiões da AWS (padrão) ou em uma única Região da AWS. Defina esse valor para a Região da AWS que a API está sendo implantada para restringir o acesso.
- **ParallelClusterFunctionRole**- Isso substitui a função do IAM que é atribuída aos recursos de implementação AWS ParallelCluster da AWS Lambda função. O parâmetro aceita o ARN de um perfil do IAM. Essa função precisa ser configurada para ter AWS Lambda como entidade principal do IAM.
- **CustomDomainName,CustomDomainCertificate, CustomDomainHostedZoneId** - Use esses parâmetros para definir um domínio personalizado para o endpoint do Amazon API Gateway. `CustomDomainName` é o nome do domínio a ser usado, `CustomDomainCertificate` é o ARN de um certificado AWS gerenciado para esse nome de domínio e `CustomDomainHostedZoneId` é o ID da zona hospedada do [Amazon Route 53](#) na qual você deseja criar registros.


Warning

Você pode definir configurações de domínio personalizadas para impor uma versão mínima do Transport Layer Security (TLS) para a API. Para obter mais informações, consulte [Escolher uma versão mínima do TLS para um domínio personalizado no API Gateway](#).

- **EnableIamAdminAccess**- Por padrão, as operações da AWS ParallelCluster API de processamento de AWS Lambda funções são configuradas com uma função do IAM que impede

qualquer acesso privilegiado do IAM (`EnableIamAdminAccess=false`). Isso faz com que a API não consiga processar operações que exijam a criação de políticas ou perfis do IAM. Por esse motivo, a criação de clusters ou imagens personalizadas só é bem-sucedida quando os perfis do IAM são fornecidos como entrada como parte da configuração do recurso.

Quando `EnableIamAdminAccess` está configurado para `true`, a API AWS ParallelCluster recebe permissões para gerenciar a criação dos perfis do IAM necessários para implantar clusters ou gerar AMIs personalizadas.

 Warning


Definir isso como `true` concede privilégios de administrador do IAM à função AWS Lambda que processa operações do AWS ParallelCluster.

Consulte [AWS ParallelCluster políticas de exemplo de usuário para gerenciar recursos do IAM](#) para obter detalhes adicionais sobre os atributos que podem ser desbloqueados ao ativar esse modo.

- `PermissionsBoundaryPolicy`- Esse parâmetro opcional aceita um ARN de política do IAM existente que será definido como limite de permissões para todas as funções do IAM criadas pela infraestrutura da API do PC e como uma condição nas permissões administrativas do IAM para que somente funções com essa política possam ser criadas pela API do PC.

Consulte [Modo `PermissionsBoundary`](#) para obter detalhes adicionais sobre as restrições impostas por esse modo.

- `CreateApiUserRole`- Por padrão, a implantação da AWS ParallelCluster API inclui a criação de uma função do IAM que é definida como a única função autorizada a invocar a API. O endpoint do Amazon API Gateway é configurado com uma política baseada em recursos para conceder permissão de invocação somente ao usuário criado. Para alterar isso, defina `CreateApiUserRole=false` e conceda acesso à API aos usuários selecionados do IAM. Para obter mais informações, consulte [Controlar o acesso para invocar uma API](#) no Guia do desenvolvedor do Amazon API Gateway.

 Warning

Quando o acesso `CreateApiUserRole=true` ao endpoint da API não é restrito pelas políticas de recursos do Amazon API Gateway, todos os perfis do IAM com permissão

`execute-api`: Invoke irrestrita podem acessar os atributos do AWS ParallelCluster. Para obter mais informações, consulte [Como controlar o acesso a uma API com as políticas de recursos do API Gateway](#) no Guia do desenvolvedor do API Gateway.

Warning

O `ParallelClusterApiUserRole` tem permissão para invocar todas as operações da API AWS ParallelCluster. Para restringir o acesso a um subconjunto de recursos da API, consulte [Controlar quem pode chamar um método de API do API Gateway com políticas do IAM](#) no Guia do desenvolvedor do API Gateway.

- IAM RoleAndPolicyPrefix - Esse parâmetro opcional aceita uma string de no máximo 10 caracteres que será usada como prefixo para as funções e políticas do IAM criadas como parte da infraestrutura da API do PC.

Atualização da API

Atualização para uma versão mais recente do AWS ParallelCluster

Opção 1: remova a API existente excluindo a pilha AWS CloudFormation correspondente e implantando a nova API conforme mostrado acima.

Opção 2: atualize a API existente executando os seguintes comandos:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.7.0
$ 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}
```

Como invocar a API AWS ParallelCluster

O endpoint Amazon API Gateway do AWS ParallelCluster é configurado com o [tipo de autorização AWS_IAM](#), e exige que todas as solicitações sejam assinadas pelo SigV4 com credenciais válidas do IAM ([referência da API: fazer solicitações http](#)).

Quando implantadas com configurações padrão, as permissões de invocação da API são concedidas somente ao usuário do IAM padrão criado com a API.

Para recuperar o ARN do usuário do IAM padrão, execute:

```
$ 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
```

Para obter credenciais temporárias para o usuário padrão do IAM, execute o AssumeRole comando [STS](#).

É possível recuperar o endpoint da API AWS ParallelCluster executando o comando a seguir.

```
$ 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
```

A API AWS ParallelCluster pode ser invocada por qualquer cliente HTTP que esteja em conformidade com as especificações da OpenAPI que podem ser encontradas aqui:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

As solicitações precisam ser assinadas com SigV4 conforme documentado [aqui](#).

No momento, não oferecemos nenhuma implementação oficial de cliente de API. No entanto, os clientes de API podem ser facilmente gerados a partir do modelo OpenAPI usando o [OpenAPI Generator](#). Depois que o cliente é gerado, a assinatura SigV4 precisa ser adicionada, se não for fornecida imediatamente.

Uma implementação de referência para um cliente da API Python pode ser encontrada no [repositório do AWS ParallelCluster](#). Para saber mais sobre como você pode usar o cliente da API Python, consulte o tutorial [Uso da API AWS ParallelCluster](#).

Para implementar mecanismos de controle de acesso mais avançados, como Amazon Cognito ou Lambda Authorizers, ou para proteger ainda mais a API com AWS WAF ou chaves de API, siga a [documentação do Amazon API Gateway](#).

Warning

Um usuário do IAM autorizado a invocar a API AWS ParallelCluster pode controlar indiretamente todos os recursos do AWS gerenciados pelo AWS ParallelCluster na Conta da AWS. Isso inclui a criação de recursos AWS que o usuário não pode controlar diretamente devido às restrições na política do IAM do usuário. Por exemplo, a criação de um cluster do AWS ParallelCluster, dependendo de sua configuração, pode incluir a implantação de instâncias do Amazon EC2, Amazon Route 53, sistemas de arquivos Amazon Elastic File System, sistemas de arquivos Amazon FSx, perfis do IAM e recursos de outros Serviços da AWS usados pelo AWS ParallelCluster que o usuário talvez não tenha controle direto.

Warning

Ao criar um cluster com o `AdditionalIamPolicies` especificado na configuração, as políticas adicionais devem corresponder a um dos seguintes padrões:

```
- !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 precisar de outras políticas adicionais, você pode realizar uma das seguintes ações:

- Editar o DefaultParallelClusterIamAdminPolicy em:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

Adicione essa política na seção ArnLike/iam:PolicyARN.

- Omitir a especificação de políticas para AdditionalIamPolicies no arquivo de configuração e adicionar manualmente as políticas à Função de Instância do AWS ParallelCluster criada no cluster.

Acessando logs e métricas das APIs

Os registros da API são publicados na Amazon CloudWatch com uma retenção de 30 dias. Para recuperar o LogGroup nome associado a uma implantação de API, execute o seguinte 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
```

Métricas, logs e logs de rastreamento [AWS X-Ray](#) do Lambda também podem ser acessados por meio do console do Lambda. Para recuperar o ARN da função do Lambda associada a uma implantação de APIs, execute os comandos a seguir:

```
$ 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
```


Conecte-se ao nó principal por meio do NICE DCV

O NICE DCV é uma tecnologia de visualização remota que permite que os usuários se conectem com segurança a aplicativos 3D que usam muitos gráficos e que são hospedados em um servidor remoto de alto desempenho. Para obter mais informações, consulte [NICE DCV](#).

O software NICE DCV é instalado automaticamente no nó principal e pode ser ativado usando a seção [Dcv](#) da configuração do [HeadNode](#).

```
HeadNode:
  Dcv:
    Enabled: true
```

Dessa forma, o AWS ParallelCluster configura `/home/<DEFAULT_AMI_USER>` no nó principal para a [pasta de armazenamento do servidor DCV](#). Para obter mais informações sobre parâmetros de configuração do NICE DCV, consulte [HeadNode / Dcv](#). Para se conectar à sessão do NICE DCV, use o comando [pcluster dcv-connect](#).

Certificado HTTPS do NICE DCV

O NICE DCV gera automaticamente um certificado autoassinado para proteger o tráfego entre o cliente do NICE DCV e o servidor do NICE DCV.

Para substituir o certificado padrão do NICE DCV autoassinado por outro certificado, primeiro conecte-se ao nó principal. Depois, copie o certificado e a chave para a pasta `/etc/dcv` antes de executar o comando [pcluster dcv-connect](#).

Para obter mais informações, consulte [Alterando o certificado TLS](#) no Guia do administrador do NICE DCV.

Licenciamento do NICE DCV

O servidor do NICE DCV não requer um servidor de licenças na execução em instâncias do Amazon EC2. No entanto, o servidor do NICE DCV deve se conectar periodicamente a um bucket do Amazon S3 para determinar se uma licença válida está disponível.

O AWS ParallelCluster adiciona automaticamente as permissões necessárias à política do IAM do nó principal. Ao usar uma política de instância do IAM personalizada, use as permissões descritas em [NICE DCV no Amazon EC2](#) no Guia do administrador do NICE DCV.

Para obter dicas de solução de problemas, consulte [Solucionar problemas no NICE DCV](#).

Usar o `pcluster update-cluster`

Na AWS ParallelCluster versão 3.x, [pcluster update-cluster](#) analisa as configurações usadas para criar o cluster atual e as configurações no arquivo de configuração em busca de problemas. Se algum problema for descoberto, ele será relatado e as etapas a serem seguidas para corrigi-los serão exibidas. Por exemplo, se a [InstanceType](#) de computação for alterada, a frota de computação deverá ser interrompida antes que uma atualização possa continuar. Esse problema é relatado quando é descoberto. Se nenhum problema de bloqueio for descoberto, o processo de atualização será iniciado e as alterações serão relatadas.

Você pode usar o [pcluster update-cluster --dryrun](#) option para ver as alterações antes de serem executadas. Para ter mais informações, consulte [Exemplos do pcluster update-cluster](#).

Para obter ajuda de solução de problemas, consulte [Solução de problemas do AWS ParallelCluster](#).

Política de atualização: definições

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Depois de alterar essa configuração, o cluster pode ser atualizado usando o [pcluster update-cluster](#).

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Depois de alterar essa configuração, o cluster não poderá ser atualizado. Você deve reverter as configurações do cluster original e criar um novo cluster com as configurações atualizadas. Você pode excluir o cluster original em uma data posterior. Use [pcluster create-cluster](#) para criar um novo cluster. Use [pcluster delete-cluster](#) para excluir o cluster original.

Política de atualização: essa configuração não é analisada durante uma atualização.

Essas configurações podem ser alteradas, e o cluster pode ser atualizado usando o [pcluster update-cluster](#).

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

Essas configurações não podem ser alteradas enquanto a frota de computação existir. A alteração deve ser revertida ou a frota de computação deve ser interrompida (usando [pcluster update-compute-fleet](#)). Depois que a frota computacional for interrompida, você poderá

atualizar o cluster ([pcluster update-cluster](#)) para ativar as alterações. Por exemplo, se você estiver usando um programador Slurm com [SlurmQueues](#) / [ComputeResources](#) / - [Name](#) / [MinCount](#) > 0, uma frota de computação será iniciada.

Política de atualização: a frota computacional e os nós de login devem ser interrompidos para que essa configuração seja alterada para uma atualização.

Essas configurações não podem ser alteradas enquanto a frota computacional existir ou se os nós de login estiverem em uso. A alteração deve ser revertida ou a frota de computação e os nós de login devem ser interrompidos (o uso da frota de computação pode ser interrompido). [pcluster update-compute-fleet](#) Depois que a frota de computação e os nós de login forem interrompidos, você poderá atualizar o cluster ([pcluster update-cluster](#)) para ativar as alterações.

Política de atualização: essa configuração não pode ser diminuída durante uma atualização.


Essas configurações podem ser alteradas, mas não podem ser diminuídas. Se essas configurações tiverem que ser diminuídas, você deve reverter as configurações do cluster original e criar um novo cluster com as configurações atualizadas. Você pode excluir o cluster original em uma data posterior. Use [pcluster create-cluster](#) para criar um novo cluster. Use [pcluster delete-cluster](#) para excluir o cluster original.

Política de atualização: se essa configuração for alterada, a atualização não será permitida. Se você forçar a atualização, o novo valor será ignorado e o valor antigo será usado.

Depois de alterar essa configuração, o cluster não poderá ser atualizado. Você deve reverter as configurações do cluster original e criar um novo cluster com as configurações atualizadas. Você pode excluir o cluster original em uma data posterior. Use [pcluster create-cluster](#) para criar um novo cluster. Use [pcluster delete-cluster](#) para excluir o cluster original.

Política de atualização: a frota de computação deve ser interrompida ou [QueueUpdateStrategy](#) para que essa configuração seja alterada para uma atualização.

Essas configurações podem ser alteradas. A frota de computação deve ser interrompida (usando [pcluster update-compute-fleet](#)) ou [QueueUpdateStrategy](#) deve ser definido. Depois que a frota de computação for interrompida ou [QueueUpdateStrategy](#) for definido, você poderá atualizar o cluster ([pcluster update-cluster](#)) para ativar as alterações.

 Note


Essa política de atualização é suportada a partir da AWS ParallelCluster versão 3.2.0.

Política de atualização: para essa configuração de valores de lista, um novo valor pode ser adicionado durante uma atualização ou a frota de computação deve ser interrompida ao remover um valor existente.

Um novo valor para essas configurações pode ser adicionado durante uma atualização. Depois de adicionar um novo valor à lista, o cluster pode ser atualizado usando ([pcluster update-cluster](#)).

Para remover um valor existente da lista, a frota de computação deve ser interrompida (usando [pcluster update-compute-fleet](#)).

Por exemplo, se você estiver usando um Slurm agendador e adicionando um novo tipo de instância a [Instances/ InstanceType](#), poderá atualizar o cluster sem interromper a frota computacional. [Para remover um tipo de instância existente de Instances/ InstanceType, a frota computacional deve ser interrompida primeiro \(usando pcluster\). update-compute-fleet](#)

 Note

Essa política de atualização é suportada a partir da AWS ParallelCluster versão 3.2.0.

Política de atualização: reduzir o tamanho de uma fila exige que a frota de computação seja interrompida ou [QueueUpdateStrategy](#) definida como TERMINATE para que essa configuração seja alterada para uma atualização.

Essas configurações podem ser alteradas, mas se a alteração reduzir o tamanho da fila, a frota de computação deverá ser interrompida (usando o [pcluster update-compute-fleet](#)) ou definida como [QueueUpdateStrategy](#) TERMINATE. Depois que a frota computacional for interrompida ou [QueueUpdateStrategy](#) definida como TERMINATE, você poderá atualizar o cluster ([pcluster update-cluster](#)) para ativar as alterações.

O TERMINATE definido ao redimensionar a capacidade do cluster só encerrará os nós do final da lista de nós e deixará intocados todos os outros nós da mesma partição.

Por exemplo, se a capacidade inicial do cluster for `MinCount = 5` e `MaxCount = 10`, os nós serão `st-[1-5]`; `dy-[1-5]`. Ao redimensionar o cluster para `MinCount = 3` e `MaxCount = 5`, a nova capacidade do cluster será composta pelos nós `st-[1-3]`; `dy-[1-2]`, que não serão afetados durante a atualização. Somente os `st-[4-5]`; `dy-[3-5]` nós serão encerrados durante a atualização.

As alterações a seguir são suportadas e não exigem que a frota de computação seja interrompida nem a [QueueUpdateStrategy](#) configuração para TERMINATE:


- Um novo [SlurmQueue](#) é adicionado
- Um novo [ComputeResource](#) é adicionado
- [MaxCount](#) é aumentado
- [MinCount](#) é aumentado e [MaxCount](#) é aumentado em pelo menos a mesma quantidade

Observação: essa política de atualização é suportada a partir da AWS ParallelCluster versão 3.9.0.

Política de atualização: para essa configuração de valores de lista, a frota de computação deve ser interrompida ou [QueueUpdateStrategy](#) deve ser configurada para agregar um novo valor; a frota de computação deve ser interrompida ao remover um valor existente.

Um novo valor para essas configurações pode ser adicionado durante uma atualização. A frota de computação deve ser interrompida (usando [pcluster update-compute-fleet](#)) ou [QueueUpdateStrategy](#) deve ser definido. Depois que a frota de computação for interrompida ou [QueueUpdateStrategy](#) for definido, você poderá atualizar o cluster ([pcluster update-cluster](#)) para ativar as alterações.

Para remover um valor existente da lista, a frota de computação deve ser interrompida (usando [pcluster update-compute-fleet](#)).

 Note

Essa política de atualização é suportada a partir da AWS ParallelCluster versão 3.3.0.

Política de atualização: todos os nós de computação devem ser interrompidos para a exclusão de um grupo de posicionamento gerenciado. A frota de computação deve ser interrompida ou [QueueUpdateStrategy](#) deve ser configurada para que essa configuração seja alterada para uma atualização.

A frota de computação deve ser interrompida (usando [pcluster update-compute-fleet](#)) para remover um grupo de posicionamento gerenciado. Se você executar uma atualização de cluster para remover um grupo de posicionamento gerenciado antes de interromper a frota de computação, uma mensagem de configuração inválida será retornada e a atualização não prosseguirá. A interrupção da frota de computação garante que nenhuma instância esteja em execução.

Exemplos do `pcluster update-cluster`

Essas configurações podem ser alteradas, mas se a alteração reduzir o tamanho da fila, a frota de computação deverá ser interrompida (usando o `pcluster update-compute-fleet`) ou definida como [QueueUpdateStrategy](#) `TERMINATE`. Depois que a frota computacional for interrompida ou [QueueUpdateStrategy](#) definida como `TERMINATE`, você poderá atualizar o cluster ([pcluster update-cluster](#)) para ativar as alterações.

- Este exemplo demonstra uma atualização com algumas alterações permitidas e a atualização é iniciada diretamente.

```
$ 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": "3.7.0",
    "clusterStatus": "UPDATE_IN_PROGRESS"
  },
  "changeSet": [
    {
      "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
      "requestedValue": [
        "sg-0cd61884c4ad11234"
      ],
      "currentValue": [
        "sg-0cd61884c4ad16341"
      ]
    }
  ]
}
```

- Este exemplo demonstra uma atualização de simulação com algumas alterações permitidas. A simulação é útil para relatar o conjunto de alterações sem iniciar a atualização.

```
$ 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"
    ]
  }
]
}

```

- Este exemplo demonstra uma atualização com algumas alterações que bloqueiam a atualização.

```

$ 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 Personalização da AMI

Há cenários em que AWS ParallelCluster é necessário criar uma AMI personalizada para. Esta seção aborda o que considerar ao criar uma AWS ParallelCluster AMI personalizada.

Você pode criar uma AWS ParallelCluster AMI personalizada usando um dos seguintes métodos:

1. Crie um [arquivo de configuração de imagem de compilação](#) e, em seguida, use a `pcluster` CLI para criar a imagem com o EC2 Image Builder. Esse processo é automatizado, repetível e oferece suporte ao monitoramento. Para obter mais informações, consulte os comandos de [imagem `pcluster`](#).

2. Crie uma instância a partir de uma AWS ParallelCluster AMI, faça login nela e faça modificações manuais. Por último, use o Amazon EC2 para criar uma nova AMI a partir da instância modificada. Esse processo leva menos tempo. No entanto, ele não é automatizado nem repetível e não oferece suporte ao uso dos comandos de monitoramento de imagem da CLI `pcluster`.

Para ter mais informações sobre esses métodos, consulte [Criar uma AMI personalizada do AWS ParallelCluster](#).

AWS ParallelCluster Considerações sobre a personalização da AMI

Independente de como você cria sua imagem personalizada, recomendamos que você realize testes preliminares de validação e inclua provisões para monitorar o status da imagem que está sendo criada.

Para criar uma AMI personalizada usando `pcluster`, você cria um [arquivo de configuração de imagem de compilação](#) com uma [Build](#) e [seçãoImage](#) que o [EC2 Image Builder](#) usa para criar sua imagem personalizada. A seção `Build` especifica o que o Image Builder precisa para criar a imagem. Isso inclui a [ParentImage](#) (imagem base) e [Components](#). Um [componente do Image Builder](#) define uma sequência de etapas necessárias para personalizar uma instância antes que uma imagem seja criada ou para testar uma instância que foi executada pela imagem criada. Para exemplos de AWS ParallelCluster componentes, consulte [AMIs personalizadas](#). A seção `Image` especifica as propriedades da imagem.

Quando chamado do `pcluster` [build-image](#) para criar uma imagem personalizada, o Image Builder usa a configuração da imagem de compilação com o AWS ParallelCluster livro de receitas para inicializar sua. AWS ParallelCluster [ParentImage](#) O Image Builder baixa componentes, executa fases de compilação e validação, cria a AMI, executa uma instância da AMI e executa testes. Quando o processo é concluído, o Image Builder produz uma nova imagem ou uma mensagem de interrupção.

Execute testes personalizados de validação de componentes

Antes de incluir um componente do Image Builder em uma configuração, teste e valide usando um dos métodos a seguir. Como o processo do Image Builder pode levar até 1 hora, recomendamos que você teste os componentes com antecedência. Isso pode economizar uma quantidade considerável de tempo.

Caso de script

Teste o script em uma instância em execução, fora do processo de criação da imagem, e verifique se o script sai com o código de saída 0.

Caso do nome do recurso da Amazon (ARN)

Teste o documento do componente em uma instância em execução, fora do processo de criação da imagem. Para obter uma lista dos requisitos, consulte [Component manager \(Gerenciador de componentes\)](#) no Guia do usuário do Image Builder.

Após a validação bem-sucedida, adicione o componente à sua configuração de imagem de compilação

Depois de verificar se o componente personalizado está funcionando, adicione-o ao [arquivo de configuração de criação de imagem](#).

Monitore o processo do Image Builder com comandos **pcluster** para auxiliar na depuração

[describe-image](#)

Use esse comando para monitorar o status de criação da imagem.

[list-image-log-streams](#)

Use esse comando para obter os IDs dos fluxos de log que você pode usar para recuperar eventos de log com [get-image-log-events](#).

[get-image-log-events](#)

Use esse comando para obter o fluxo de log dos eventos do processo de criação de imagem.

Por exemplo, você pode iniciar eventos da imagem usando o seguinte comando.

```
$ 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](#)

Use esse comando para recuperar eventos da pilha de imagens para a pilha criada pelo Image Builder.

[export-image-logs](#)

Use este comando para salvar registros de imagens.

Para obter mais informações sobre AWS ParallelCluster registros e a Amazon CloudWatch, consulte [Logs de imagem de compilação do Amazon CloudWatch Logs Painel do Amazon CloudWatch](#) e.

Outras considerações

Novos AWS ParallelCluster lançamentos e AMIs personalizadas

Se você criar e usar uma AMI personalizada, será necessário repetir as etapas usadas para criar sua AMI personalizada com cada versão nova do AWS ParallelCluster .

Ações de bootstrap personalizadas

Revise a [Ações de bootstrap personalizadas](#) seção para determinar se as modificações que você deseja fazer podem ser programadas e suportadas em AWS ParallelCluster versões futuras.

Uso de AMIs personalizadas

Você pode especificar AMIs personalizadas na configuração do cluster nas seções [Image](#) / [CustomAmi](#) e [Scheduling](#) / [SlurmQueues](#) / - [Name](#) / [Image](#) / [CustomAmi](#).

Para solucionar problemas de avisos de validação da AMI personalizada, consulte [Solução de problemas de AMI personalizada](#).

Iniciar instâncias com ODCR (On-Demand Capacity Reservations)

Com as [Reservas de capacidade sob demanda](#) (ODCRs), você pode ter uma reserva de capacidade para instâncias do Amazon EC2 do seu cluster em uma zona de disponibilidade específica. Dessa forma, você pode criar e gerenciar Reservas de Capacidade independentemente dos descontos de faturamento oferecidos por [Savings Plans](#) ou [Instâncias reservadas regionais](#).

Você pode configurar reservas de capacidade sob demanda (ODCRs) open ou targeted. As ODCRs Abertas abrangem todas as instâncias que correspondam aos atributos do ODCR. Esses atributos são tipo de instância, plataforma e zona de disponibilidade. Você deve definir explicitamente as ODCRs Direcionadas na configuração do cluster. Para determinar se uma ODCR é open ou targeted, execute o comando [describe-capacity-reservation](#) do EC2 da AWS CLI.

Você também pode criar uma ODCR em um grupo de posicionamento de cluster é chamado de [cluster placement group on-demand capacity reservation \(CPG ODCR\)](#).

Várias ODCRs podem ser agrupados em um grupo de recursos. Isso pode ser definido no arquivo de configuração do cluster. Para obter mais informações sobre grupos de recursos, consulte [O que são grupos de recursos?](#) no Guia do usuário de Grupos de Recursos e Tags.

Usando o ODCR com o AWS ParallelCluster

O AWS ParallelCluster suporta ODCRs abertos. Quando você usa um ODCR aberto, não precisa especificar nada no AWS ParallelCluster. As instâncias são selecionadas automaticamente para o cluster. É possível especificar um grupo de posicionamento existente ou deixar que o AWS ParallelCluster crie um novo para você.

ODCR na configuração do cluster

A partir do AWS ParallelCluster versão 3.3.0, você pode definir ODCRs no arquivo de configuração do cluster, sem precisar especificar substituições de instâncias de execução do EC2.

Comece criando [reservas de capacidade](#) e [grupos de recursos](#) usando os métodos descritos na documentação vinculada para cada um. Você deve usar os métodos AWS CLI para criar grupos de reserva de capacidade. Se você usar o AWS Management Console, só poderá criar grupos de recursos baseados em tags ou em pilhas. Os grupos de recursos baseados em tags e baseados em pilhas não são compatíveis com o AWS ParallelCluster ou a AWS CLI ao executar instâncias com reservas de capacidade.

Depois que as reservas de capacidade e os grupos de recursos forem criados, especifique-os em [SlurmQueues / CapacityReservationTarget](#) ou [SlurmQueues / ComputeResources / CapacityReservationTarget](#), conforme mostrado no exemplo de configuração de cluster a seguir. Substitua os *valores* destacados em vermelho pelos valores válidos.

```
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 / NÃO RECOMENDADO – ODCR direcionado com substituições de instâncias do EC2

Warning

- A partir do AWS ParallelCluster versão 3.3.0, não recomendamos esse método. Esta seção permanece como referência para implementações usando versões anteriores.
- Esse método não é compatível com a alocação de vários tipos de instâncias com o Slurm.

Compatibilidade para ODCRs `targeted` adicionado no AWS ParallelCluster versão 3.1.1. Nessa versão foi introduzido um mecanismo que substitui os parâmetros `RunInstances` do EC2 e transmite informações sobre a reserva a ser usada para cada recurso de computação configurado em AWS ParallelCluster. Esse mecanismo é compatível com ODCRs `targeted`. No entanto, ao usar ODCRs `targeted`, você deve especificar a configuração de substituição de `run-instances`. Os ODCRs Direcionados devem ser definidos explicitamente no comando [run-instances](#) do EC2 da AWS CLI. Para determinar se uma ODCR é `open` ou `targeted`, execute o comando [describe-capacity-reservation](#) do EC2 da AWS CLI.

Várias ODCRs podem ser agrupados em um grupo de recursos. Isso pode ser usado na substituição de instâncias de execução para atingir vários ODCRs ao mesmo tempo.

Se você estiver usando um ODCR `targeted`, poderá especificar um grupo de posicionamento. No entanto, você também precisa especificar uma configuração de substituição do `run-instances`.

Suponha que o AWS tenha criado um ODCR `targeted` para você ou que você tenha um conjunto específico de instâncias reservadas. Então, você não pode especificar um grupo de posicionamento. As regras configuradas por AWS podem entrar em conflito com a configuração do grupo de

posicionamento. Portanto, se for necessário um grupo de posicionamento para seu aplicativo, use um [CPG ODCR](#). Em ambos os casos, você também deve especificar a configuração de substituição `run-instances`.

Se você estiver usando um CPG ODCR, deverá especificar a configuração de substituição `run-instances` e especificar o mesmo grupo de posicionamento na configuração do cluster.

Usar instâncias reservadas com o AWS ParallelCluster

As instâncias reservadas [são diferentes](#) das reservas de capacidade (ODCR). Há [dois tipos](#) de instâncias reservadas. Uma Instância reservada Regional não reserva capacidade. Uma Instância reservada por zona reserva capacidade na zona de disponibilidade especificada.

Se você tiver instâncias reservadas regionais, não há reserva de capacidade e você pode obter erros de capacidade insuficientes. Se você tem instâncias reservadas por zona, você tem reserva de capacidade, mas não há parâmetros de API `run-instances` que você possa usar para especificá-las.

As instâncias reservadas são compatíveis com qualquer versão do AWS ParallelCluster. Você não precisa especificar nada no AWS ParallelCluster, e as instâncias serão selecionadas automaticamente.

Ao usar instâncias reservadas por zona, você pode evitar possíveis erros de capacidade insuficiente omitindo a especificação do grupo de posicionamento na configuração do cluster.

OBSOLETO/NÃO RECOMENDADO – Uso da personalização **RunInstances** no AWS ParallelCluster 3 para reservas de capacidade **targeted** sob demanda (ODCRs)

Warning

- A partir do AWS ParallelCluster versão 3.3.0, não recomendamos esse método. Esta seção permanece como referência para implementações usando versões anteriores.
- Esse método não é compatível com a alocação de vários tipos de instâncias com o Slurm.

Você pode substituir os parâmetros `RunInstances` do EC2 para cada recurso de computação configurado em uma fila de cluster. Para fazer isso, crie o arquivo `/opt/slurm/etc/pcluster/run_instances_overrides.json` no nó principal do cluster com o seguinte conteúdo de trecho de código:

- `${queue_name}` é o nome da fila à qual você deseja aplicar substituições.
- `${compute_resource_name}` é o recurso de computação ao qual você deseja aplicar substituições.
- `${overrides}` é um objeto JSON arbitrário que contém uma lista de substituições `RunInstances` a serem usadas na combinação específica de fila e tipo de instância. A sintaxe das substituições precisa seguir as mesmas especificações documentadas em uma chamada [run_instances boto3](#).

```
{
  "${queue_name}": {
    "${compute_resource_name}": {
      ${overrides}
    },
    ...
  },
  ...
}
```

Por exemplo, o JSON a seguir configura o grupo ODCR `group_arn` para ser usado em instâncias `p4d.24xlarge` configuradas em `my-queue` e `my-compute-resource`.

```
{
  "my-queue": {
    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationResourceGroupArn": "group_arn"
        }
      }
    }
  }
}
```

Depois que esse arquivo JSON é gerado, os daemons AWS ParallelCluster responsáveis pelo dimensionamento do cluster usam automaticamente a configuração de substituição para as execuções de instâncias. Para confirmar se os parâmetros especificados estão sendo usados para provisionamento de instâncias, consulte os seguintes arquivos de log:

- `/var/log/parallelcluster/clustermgtd` (para capacidade estática)

- `/var/log/parallelcluster/slurm_resume.log` (para capacidade dinâmica)

Se os parâmetros estiverem corretos, você encontrará uma entrada de log que contém o seguinte:

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

OBSOLETO/NÃO RECOMENDADO - Criar um cluster com reservas de capacidade **targeted** sob demanda (ODCRs)

Warning

- A partir do AWS ParallelCluster versão 3.3.0, não recomendamos esse método. Esta seção permanece como referência para implementações usando versões anteriores.
- Esse método não é compatível com [Alocação a vários tipos de instância com o Slurm](#).

1. Crie um grupo de recursos, para agrupar a capacidade.

```
$ 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

Um grupo de recursos não oferece suporte a recursos compartilhados por outras contas. Se o ODCR de destino for compartilhado por outra conta, você não precisará criar um grupo de recursos. Use `CapacityReservationId` em vez de um grupo de recursos na etapa 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

```

Adicione reservas de capacidade ao grupo de recursos. Toda vez que você criar um novo ODCR, adicione-o à Reserva de Grupo. Substitua *ACCOUNT_ID* pelo ID da sua conta, *PLACEHOLDER_CAPACITY_RESERVATION* pelo seu ID de reserva de capacidade e *REGION_ID* pelo seu ID da Região da AWS (por exemplo, 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

```

Crie um documento de política no computador local. Substitua *ACCOUNT_ID* pelo ID da sua conta e *REGION_ID* pelo seu ID da Região da AWS (por exemplo, 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. Crie a política do IAM em sua Conta da AWS usando o arquivo json que você criou.

```
$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-document file://policy.json
```

3. Crie o seguinte script de pós-instalação localmente na instância e o nomeie como **postinstall.sh**.

Substitua **ACCOUNT_ID** pelo seu ID da Conta da AWS e **REGION_ID** pelo seu ID da Região da AWS (por exemplo, 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
```

Faça upload do arquivo para um bucket do Amazon S3. Substitua **S3_NAME_BUCKET** pelo nome específico do bucket do S3.

```
$ aws s3 mb s3://S3_NAME_BUCKET
aws s3 cp postinstall.sh s3://S3_NAME_BUCKET/postinstall.sh
```

4. Crie a configuração do cluster local, substituindo os espaços reservados com valores próprios.

```
Region: REGION_ID
Image:
  Os: alinux2
HeadNode:
  InstanceType: c5.2xlarge
```

```
Ssh:
  KeyName: YOUR_SSH_KEY
Iam:
  S3Access:
    - BucketName: S3_NAME_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://S3_NAME_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. Crie um cluster.

Use o seguinte comando para criar o cluster: Substitua *cluster-config.yaml* pelo nome do arquivo de configuração, *cluster-dl* pelo nome do seu cluster e *REGION_ID* pelo ID da região (por exemplo, us-east-1).

```
$ pcluster create-cluster --cluster-configuration cluster-config.yaml --cluster-name cluster-dl --region REGION_ID
```

Depois que o cluster é criado, o script de pós-instalação é executado no nó principal. O script cria o arquivo `run_instances_overrides.json` e substitui as chamadas enviadas a `RunInstances` para forçar o provisionamento da partição a passar pela reserva de capacidade sob demanda.

Os daemons AWS ParallelCluster responsáveis pelo dimensionamento do cluster usam automaticamente essa configuração para novas instâncias que são iniciadas. Para confirmar se os parâmetros especificados estão sendo usados para provisionamento de instâncias, consulte os seguintes arquivos de log:

- `/var/log/parallelcluster/clustermgtd` (para capacidade estática - [MinCount](#) > 0)
- `/var/log/parallelcluster/slurm_resume.log` (para capacidade dinâmica)

Se os parâmetros estiverem corretos, você encontrará uma entrada de log que contém o seguinte:

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

Atualizando substituições `RunInstances`

Você pode atualizar a configuração JSON gerada a qualquer momento sem interromper a frota de computação. Depois que as alterações forem aplicadas, todas as novas instâncias serão iniciadas com a configuração atualizada. Se você precisar aplicar a configuração atualizada aos nós em execução, recicle os nós forçando o encerramento da instância e aguarde até AWS ParallelCluster fazer a substituição desses nós. Você pode fazer isso encerrando a instância no console do EC2 ou da AWS CLI, ou configurando os nós do Slurm em um estado DOWN ou DRAIN.

Use o comando a seguir para definir o nó Slurm para DOWN ou 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
```

Aplicação de patches de AMI e substituição de instâncias do EC2

Para garantir que todos os nós de computação do cluster iniciados dinamicamente se comportem de maneira consistente, o AWS ParallelCluster desativa as atualizações automáticas do sistema operacional da instância do cluster. Além disso, um conjunto específico de AMIs do AWS ParallelCluster é criado para cada versão do AWS ParallelCluster e sua CLI associada. Esse conjunto específico de AMIs permanece inalterado e só é compatível com a versão do AWS ParallelCluster para a qual foram criadas. AWS ParallelCluster AMIs para as versões lançadas não são atualizadas.

No entanto, devido a problemas de segurança emergentes, talvez os clientes queiram adicionar correções a essas AMIs e depois atualizar seus clusters com a AMI corrigida. Isso se alinha ao [Modelo de Responsabilidade Compartilhada do AWS ParallelCluster](#).

Para ver o conjunto específico de AMIs do AWS ParallelCluster compatível com a versão da CLI do AWS ParallelCluster que você está usando atualmente, execute:

```
$ pcluster version
$ pcluster list-official-images
```

O nó principal do AWS ParallelCluster é uma instância estática e você pode atualizá-lo manualmente. A reinicialização e reboot do nó principal são totalmente suportadas a partir do AWS ParallelCluster versão 3.0.0.

Se suas instâncias tiverem armazenamentos de instâncias efêmeros, lembre-se de salvar os dados do armazenamento de instância antes das atualizações manuais. Para obter mais informações, consulte a configuração de cluster [HeadNode](#) / [LocalStorage](#) / [EphemeralVolume](#) e [os tipos de instância com volumes de armazenamento de instâncias](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Os nós de computação são instâncias efêmeras. Por padrão, você só pode acessá-los a partir do nó principal. A partir do AWS ParallelCluster versão 3.0.0, você pode atualizar a AMI associada às instâncias de computação modificando o parâmetro [Scheduling](#) / [SlurmQueues](#) / [Image](#) / [CustomAmi](#) e executando o comando [pcluster update-cluster](#), depois de interromper a frota de computação com [pcluster update-compute-fleet](#):

```
$ pcluster update-compute-fleet-status --status STOP_REQUESTED
```

É possível automatizar a criação de uma AMI personalizada atualizada para os nós de computação usando um dos seguintes métodos:

- Use o comando `pcluster build-image` com uma [Compilação](#) atualizada / [ParentImage](#).
- Execute a compilação com `Build / UpdateOsPackages / Enabled:true`.

Atualização ou substituição da instância do nó principal

Em algumas circunstâncias, talvez seja necessário reiniciar ou fazer reboot do nó principal. Por exemplo, isso é necessário quando você atualiza manualmente o sistema operacional ou quando há uma [desativação programada da instância AWS](#) que impõe a reinicialização da instância do nó principal.

Se sua instância não tiver unidades efêmeras, você poderá interrompê-la e reiniciá-la a qualquer momento. No caso de uma desativação programada, ao iniciar a instância parada, ela será migrada para usar o novo hardware.

Da mesma forma, você pode parar e iniciar manualmente uma instância que não tenha armazenamentos de instâncias. Para esse caso e para outros casos de instâncias sem volumes efêmeros, continue para [Interromper e iniciar o nó principal de um cluster](#).

Se sua instância tiver unidades efêmeras e estiver interrompida, os dados no armazenamento de instâncias serão perdidos. Você pode determinar se o tipo de instância usado para o nó principal tem armazenamento de instâncias na tabela encontrada em [Volumes de armazenamento de instâncias](#).

Salvar dados de unidades efêmeras

A partir do AWS ParallelCluster versão 3.0.0, a reinicialização e reboot do nó principal é totalmente compatível com todo tipo de instância. No entanto, se as instâncias tiverem uma unidade efêmera, os dados serão perdidos. Siga as próximas etapas para preservar seus dados antes da reinicialização ou reboot do nó principal.

Para verificar se você tem dados que precisam ser preservados, visualize o conteúdo na pasta [EphemeralVolume](#) / [MountDir](#) (/scratch por padrão).

Você pode transferir os dados para o volume raiz ou para os sistemas de armazenamento compartilhado conectados ao cluster, como o Amazon FSx, Amazon EFS ou Amazon EBS. Observe que a transferência de dados para o armazenamento remoto pode incorrer em custos adicionais.

Depois de salvar os dados, continue para [Interromper e iniciar o nó principal de um cluster](#).

Interromper e iniciar o nó principal de um cluster

1. Verifique se não há nenhum trabalho em execução no cluster.

Ao usar um programador Slurm:

- Se a opção `sbatch --no-requeue` não for especificada, os trabalhos em execução serão enfileirados novamente.
- Se a opção `--no-requeue` for especificada, os trabalhos em execução falharão.

2. Solicite a interrupção da frota de computação em cluster:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOP_REQUESTED",
  ...
}
```

3. Espere até que o status da frota de computação seja STOPPED:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOPPED",
  ...
}
```

4. Para atualizações manuais com a reinicialização do sistema operacional ou a reinicialização da instância, você pode usar AWS Management Console o AWS CLI. Veja a seguir um exemplo de uso do 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. Iniciar a frota de computação em cluster:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
  "status": "START_REQUESTED",
  ...
}
```


Sistema operacional

AWS ParallelCluster suporta Amazon Linux 2, CentOS 7, Ubuntu 22.04, Ubuntu 2004, Red Hat Enterprise Linux 8 (RHEL8), Rocky 8, Red Hat Enterprise Linux 9 (RHEL9), Rocky 9. AWS ParallelCluster oferece AMIs pré-criadas para sistemas operacionais selecionados. Para obter mais detalhes sobre as AMIs fornecidas, AWS ParallelCluster consulte. [Seção Image](#)

Considerações sobre o sistema operacional

Ubuntu 22.04

O Ubuntu 2204 requer chaves mais seguras para ssh e não suporta chaves RSA por padrão. Gere uma chave ed25519 e use-a para a criação do cluster.

O Ubuntu 2204 não pode ser atualizado para o kernel mais recente porque não há cliente Fsx para esse kernel.

RHEL 8

RedHat O Enterprise Linux 8.7 (rhel8) é adicionado a partir da versão 3.6.0. AWS ParallelCluster Se você configurar seu cluster para usar rhel8, o custo sob demanda para qualquer tipo de instância será maior do que quando você configura seu cluster para usar outros sistemas operacionais compatíveis.

Para obter mais informações sobre preços, consulte [Preços sob demanda](#) e [Como o Red Hat Enterprise Linux no Amazon EC2 é oferecido e definido seu preço?](#).

Rocky 8

AWS ParallelCluster O 3.8.0 oferece suporte ao Rocky Linux 8, mas as AMIs pré-construídas do Rocky Linux 8 (para arquiteturas x86 e ARM) não estão disponíveis. AWS ParallelCluster O 3.8.0 suporta a criação de clusters com o Rocky Linux 8 usando AMIs personalizadas usando a propriedade. [CustomAmi](#) Para obter mais informações sobre a criação de AMIs personalizadas, consulte. [AWS ParallelCluster Personalização da AMI](#)

[Para criar sua AMI personalizada a partir de uma AMI básica do Rocky Linux 8, considere assinar as AMIs do Rocky Linux 8 disponíveis no Marketplace. AWS](#) Certifique-se de revisar os preços e os custos de assinatura das AMIs do Rocky Linux 8 no Marketplace AWS . Como alternativa, você também pode usar as [AMIs oficiais do Rocky Linux 8 como sua AMI](#) básica.

Centos 7

O [Gdrcopy](#) removeu o Centos7 de sua matriz de suporte do sistema operacional. Isso significa que o gdrcopy 2.3.1 é a versão mais recente que suporta esse sistema operacional. Você deve fixar as versões NVIDIA e gdrcopy para o Centos7, porque as versões mais recentes do driver de código aberto da NVIDIA (OpenRM, ou seja, 535.129.03+) são incompatíveis com essa versão do gdrcopy. A partir da ParallelCluster versão 3.8.0, nossas AMIs Centos7 oficiais serão lançadas com o gdrcopy 2.3.1 e o driver NVIDIA 535.129.03.

Rocky 9

AWS ParallelCluster O 3.9.0 oferece suporte ao Rocky Linux 9, mas as AMIs pré-construídas do Rocky Linux 9 (para arquiteturas x86 e ARM) não estão disponíveis. AWS ParallelCluster O 3.9.0 suporta a criação de clusters com o Rocky Linux 9 usando AMIs personalizadas usando a propriedade [CustomAmi](#). Para obter mais informações sobre a criação de AMIs personalizadas, consulte [Personalização de AWS ParallelCluster AMIs](#). Para criar sua AMI personalizada a partir de uma AMI básica do Rocky Linux 9, você também pode usar as [AMIs oficiais do Rocky Linux 9 como sua AMI](#) básica. A compilação personalizada da AMI Rocky Linux 9 pode falhar se a AMI básica não tiver o kernel mais recente. Para atualizar o kernel antes de criar a AMI:

- [Execute uma instância usando um ID de AMI rocky9 aqui: https://rockylinux.org/cloud-images/](https://rockylinux.org/cloud-images/)
- insira ssh na instância e execute o seguinte comando: `sudo yum -y update`
- Crie uma imagem da instância para usar como ParentImage

Referência para AWS ParallelCluster

Tópicos

- [Comandos da CLI do AWS ParallelCluster](#)
- [Arquivos de configuração](#)
- [Referência da API do AWS ParallelCluster](#)
- [API da biblioteca Python do AWS ParallelCluster](#)

Comandos da CLI do AWS ParallelCluster

`pcluster` é o comando principal da CLI do AWS ParallelCluster. Você usa `pcluster` para iniciar e gerenciar clusters de HPC na Nuvem AWS e para criar e gerenciar imagens AMI personalizadas.

O `pcluster3-config-converter` é usado para converter configurações de cluster no formato do AWS ParallelCluster versão 2 para o formato do AWS ParallelCluster versão 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]
```

Tópicos

- [pcluster](#)
- [pcluster3-config-converter](#)

pcluster

`pcluster` é o comando principal da AWS ParallelCluster CLI. Use a `pcluster` para executar e gerenciar clusters HPC na Nuvem AWS.

O `pcluster` grava logs de seus comandos em arquivos `pcluster.log.#` em `/home/user/.parallelcluster/`. Para ter mais informações, consulte [Logs da CLI pcluster](#).

Para usar o `pcluster`, você tem um perfil do IAM com as [permissões](#) necessárias para executar.

```
pcluster [-h]
```

Argumentos

`pcluster` *command*

Possíveis opções: [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](#)

Subcomandos:

Tópicos

- [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

Crie uma AWS ParallelCluster imagem personalizada na região especificada.

```
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}]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster build-image`.

--image-configuration, -c *IMAGE_CONFIGURATION*

Especifica o arquivo de configuração da imagem como um documento YAML.

--image-id, -i *IMAGE_ID*

Especifica o id da imagem que será criada.

--debug

Ative o registro da depuração.

--dryrun *DRYRUN*

Quando `true`, o comando executa a validação sem criar recursos. Use esse parâmetro para validar a configuração da imagem. (Padronizado como `false`).

--query *QUERY*

Consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a configuração [Região](#) no arquivo de configuração da imagem, a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--rollback-on-failure *ROLLBACK_ON_FAILURE*

Se definido como `true`, inicia automaticamente uma reversão da pilha de imagens em caso de falhas. (Padronizado como `false`.)

--suppress-validators *SUPPRESS_VALIDATORS* [*SUPPRESS_VALIDATORS ...*]

Identifica um ou mais validadores de configuração a serem suprimidos.

Formato: `(ALL|type:[A-Za-z0-9]+)`

--validation-failure-level {INFO,WARNING,ERROR}

O nível mínimo de validação que faz com que a criação de imagem falhe. (Padronizado como `ERROR`.)

Exemplo usando a AWS ParallelCluster versão 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

O `pcluster build-image` usa a VPC padrão. Se a VPC padrão tiver sido excluída, talvez usando AWS Control Tower nossa AWS Landing Zone, a ID da sub-rede deverá ser especificada no arquivo de configuração da imagem. Para obter mais informações, consulte [SubnetId](#).

pcluster configure

Inicia um assistente de configuração interativo para a AWS ParallelCluster versão 3. Para ter mais informações, consulte [Configure e crie um cluster com a interface de linha de AWS ParallelCluster comando](#).

```
pcluster configure [-h]  
                --config CONFIG  
                [--debug]  
                [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster configure`.

--config *CONFIG*

Caminho de saída do arquivo de configuração gerado.

--debug

Ative o registro da depuração.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. A Região deve ser especificada usando a configuração [Região](#) no arquivo de configuração da imagem, a variável de ambiente `AWS_DEFAULT_REGION`, a configuração `region` na seção `[default]` do arquivo `~/.aws/config` ou o parâmetro `--region`.

pcluster create-cluster

Cria um 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}]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster create-cluster`.

--cluster-configuration, -c *CLUSTER_CONFIGURATION*

Especifica o arquivo de configuração do cluster YAML.

--cluster-name, -n *CLUSTER_NAME*

O nome do novo cluster a ser criado.

O nome deve iniciar com um caractere alfabético. O nome do pode ter até 60 caracteres. Se a contabilidade do Slurm estiver ativada, o nome poderá ter até 40 caracteres.

Caracteres válidos: a-z, A-Z, 0-9 e - (hífen).

--debug

Habilitar log de depuração

--dryrun *DRYRUN*

Quando `true`, o comando executa a validação sem criar recursos. Você pode usar isso para validar a configuração do cluster. (Padronizado como `false`.)

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a [Region](#) configuração no arquivo de configuração do cluster, a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--rollback-on-failure *ROLLBACK_ON_FAILURE*

Se definido como `true`, inicia automaticamente uma reversão da pilha de cluster em caso de falhas. (Padronizado como `true`.)

--suppress-validators *SUPPRESS_VALIDATORS* [*SUPPRESS_VALIDATORS* ...]

Identifica um ou mais validadores de configuração a serem suprimidos.

Formato: `(ALL|tipo:[A-Za-z0-9]+)`

--validation-failure-level {INFO,WARNING,ERROR}

O nível mínimo de validação que faz com que a criação de imagem falhe. (Padronizado como `ERROR`.)

Exemplo usando a AWS ParallelCluster versão 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

Permite conectar-se ao nó principal por meio de uma sessão interativa usando o NICE DCV.

```
pcluster dcv-connect [-h]
                    --cluster-name CLUSTER_NAME
                    [--debug]
                    [--key-path KEY_PATH]
                    [--region REGION]
                    [--show-url]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster dcv-connect`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--key-path *KEY_PATH*

Especifica o caminho da chave SSH a ser usada para a conexão.

--region, -r *REGION*

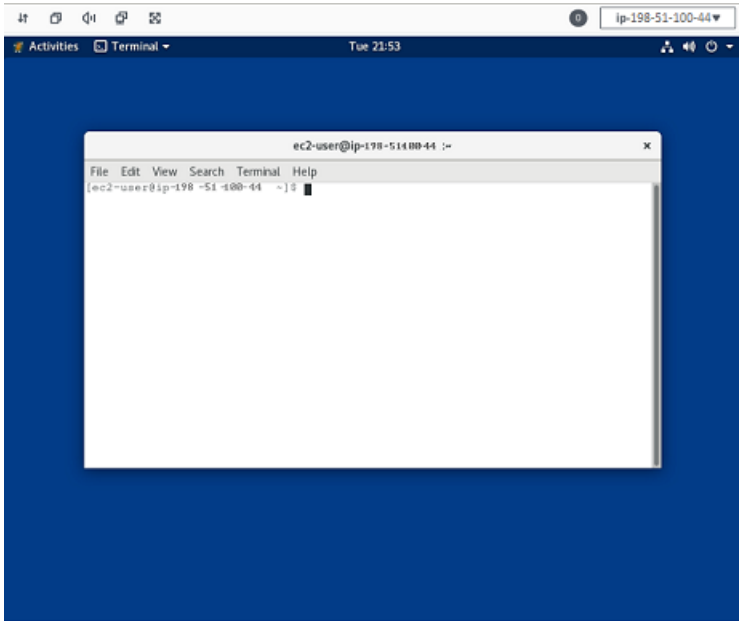
Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--show-url

Imprime o URL que seria usado para a conexão DCV e sai.

Exemplo usando a AWS ParallelCluster versão 3.1.4:

```
$ pcluster dcv-connect -n cluster-3Dcv -r us-east-1 --key-path /home/user/.ssh/key.pem
```



pcluster delete-cluster

Inicia a exclusão de um cluster.

```
pcluster delete-cluster [-h]
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster delete-cluster`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. A região deve ser especificada usando a variável de ambiente `AWS_DEFAULT_REGION`, a configuração `region` na seção `[default]` do arquivo `~/.aws/config` ou o parâmetro `--region`.

Exemplo usando a AWS ParallelCluster versão 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

Inicia o encerramento forçado de todos os nós de computação do cluster. Isso não funciona com AWS Batch clusters.

```
pcluster delete-cluster-instances [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--force FORCE]
    [--query QUERY]
```

```
[--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster delete-cluster-instances`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--force *FORCE*

Quando `true`, força a exclusão ignorando os erros de validação. (Padronizado como `false`.)

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

```
$ pcluster delete-cluster-instances -n cluster-v3
```

pcluster delete-image

Inicia a exclusão da AWS ParallelCluster imagem personalizada.

```
pcluster delete-image [-h]
  --image-id IMAGE_ID
  [--debug]
  [--force FORCE]
  [--query QUERY]
  [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster delete-image`.

--image-id, -i *IMAGE_ID*

Especifica o id da imagem que será excluída.

--debug

Habilitar log de depuração

--force *FORCE*

Quando `true`, força a exclusão caso haja instâncias usando a AMI ou se a AMI for compartilhada. (Padronizado como `false`.)

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 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

Obter informações detalhadas sobre um cluster.

```
pcluster describe-cluster [-h]
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster describe-cluster`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplos usando a AWS ParallelCluster versão 3.1.4:

Descrever os detalhes do 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",
  "address": "8af2145440569xyz.us-east-1.amazonaws.com",
  "scheme": "internet-facing|internal",
  "healthyNodes": 3,
  "unhealthyNodes": 0
},
"version": "3.1.4",
"clusterConfiguration": {
  "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
},
"tags": [
  {
    "value": "3.1.4",
    "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"
}

```

Use `describe-cluster` para recuperar a configuração do 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

Descreve as instâncias em um 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]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster describe-cluster-instances`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--node-type {HeadNode,ComputeNode,LoginNode}

Especifica os tipos de nós a serem listados. Os valores com suporte HeadNode, ComputeNode e LoginNode. Se esse parâmetro não for especificado, as instâncias HeadNode, ComputeNode e LoginNode serão descritas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--queue-name *QUEUE_NAME*

Especifica o nome da fila a ser definido. Se esse parâmetro não for especificado, as instâncias em todas as filas serão descritas.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de AWS_DEFAULT_REGION ambiente, a region configuração na [default] seção do ~/.aws/config arquivo ou o --region parâmetro.

Exemplo usando a AWS ParallelCluster versão 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"
  }
]
}
```

pcluster describe-compute-fleet

Descreve o status da frota de computação.

```
pcluster describe-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--query QUERY]
    [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster describe-compute-fleet`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 3.1.4:

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
```

```
"status": "RUNNING",
"lastStatusUpdatedTime": "2022-07-12T17:24:26.000Z"
}
```

pcluster describe-image

Obtém informações detalhadas sobre uma imagem.

```
pcluster describe-image [-h]
                        --image-id IMAGE_ID
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster describe-image`.

--image-id, -i *IMAGE_ID*

Especifica a ID da imagem.

--debug

Habilitar log de depuração

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 3.1.2:

```
$ pcluster describe-image --image-id custom-linux2-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:s3_bucket"
  },
  {
    "value": "custom-alinux2-image",
    "key": "parallelcluster:image_name"
  },
  {
    "value": "available",
    "key": "parallelcluster:build_status"
  },
  {
    "value": "s3://parallelcluster-1234abcd5678efgh-v1-do-not-delete/
parallelcluster/3.1.2/images/custom-alinux2-image-1234abcd5678efgh/configs/image-
config.yaml",
    "key": "parallelcluster:build_config"
  },
  {
    "value": "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/3.7.0/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

Exporta os logs do cluster para um arquivamento local `tar.gz` passando por um bucket do Amazon S3.

```
pcluster export-cluster-logs [-h]
    --bucket BUCKET_NAME
    --cluster-name CLUSTER_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]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster export-cluster-logs`.

--bucket *BUCKET_NAME*

Especifica o nome do bucket do Amazon S3 para o qual exportar dados de logs de cluster. Deve estar na mesma região do cluster

Note

Você deve adicionar permissões à política de bucket do Amazon S3 para conceder CloudWatch acesso. Para obter mais informações, consulte [Definir permissões em um bucket do Amazon S3 no Guia](#) do usuário de CloudWatch registros.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--bucket-prefix *BUCKET_PREFIX*

Especifica o caminho no bucket do Amazon S3 para o qual os dados de log exportados devem ser armazenados.

Por padrão, o prefixo do bucket é:

```
cluster-name-logs-202209061743.tar.gz
```

202209061743 é a hora atual no formato %Y%m%d%H%M.

--debug

Habilitar log de depuração

--end-time *END_TIME*

Especifica o final do intervalo de tempo para coletar eventos de log, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo 2021-01-01T20:00:00Z). Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos. Elementos de tempo (por exemplo, minutos e segundos) podem ser omitidos. O valor padrão é o horário atual.

--filters *FILTER* [*FILTER* ...]

Especifica filtros para o log. Formato: Name=a, Values=1 Name=b, Values=2, 3. Filtros compatíveis para:

private-dns-name

Especifica a forma abreviada do nome DNS privado da instância (por exemplo, ip-10-0-0-101).

node-type

Especifica o tipo de nó, o único valor aceito para esse filtro é HeadNode.

--keep-s3-objects *KEEP_S3_OBJECTS*

Se for true, as exportações de objetos exportados para o Amazon S3 serão mantidas. (Padronizado como false.)

--output-file *OUTPUT_FILE*

Especifica o caminho do arquivo onde será salvo o arquivamento de log. Se isso for fornecido, os logs serão salvos localmente. Caso contrário, eles serão enviados para o Amazon S3 com o URL retornado na saída. O padrão é fazer upload para o Amazon S3.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de AWS_DEFAULT_REGION ambiente, a region configuração na [default] seção do ~/.aws/config arquivo ou o --region parâmetro.

--start-time *START_TIME*

Especifica o início do intervalo de tempo, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo 2021-01-01T20:00:00Z). Eventos de log com uma marca de data/hora igual ou posterior a esse horário estão incluídos. Se não for especificado, o padrão é a hora em que o cluster foi criado.

Exemplo usando a AWS ParallelCluster versão 3.1.4:

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
{
  "url": "https://cluster-v3-bucket..."
}
```

pcluster export-image-logs

Exporta os logs da pilha do construtor de imagens para um arquivo tar.gz local passando por um bucket do Amazon S3.

```
pcluster export-image-logs [-h]
    --bucket BUCKET
    --image-id IMAGE_ID
    [--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]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster export-image-logs`.

--bucket *BUCKET_NAME*

Especifica o nome do bucket do Amazon S3 para o qual exportar logs criação de imagem. Deve estar na mesma região da imagem.

Note

Você deve adicionar permissões à política de bucket do Amazon S3 para conceder CloudWatch acesso. Para obter mais informações, consulte [Definir permissões em um bucket do Amazon S3 no Guia](#) do usuário de CloudWatch registros.

--image-id, -i *IMAGE_ID*

O ID da imagem cujos logs serão exportados.

--bucket-prefix *BUCKET_PREFIX*

Especifica o caminho no bucket do Amazon S3 para o qual os dados de log exportados devem ser armazenados.

Por padrão, o prefixo do bucket é:

```
ami-id-logs-202209061743.tar.gz
```

202209061743 é a hora atual no formato %Y%m%d%H%M.

--debug

Habilitar log de depuração

--end-time **END_TIME**

Especifica o final do intervalo de tempo para coletar eventos de log, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo 2021-01-01T20:00:00Z). Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos. Elementos de tempo (por exemplo, minutos e segundos) podem ser omitidos. O valor padrão é o horário atual.

--keep-s3-objects **KEEP_S3_OBJECTS**

Se for `true`, as exportações de objetos exportados para o Amazon S3 serão mantidas. (Padronizado como `false`.)

--output-file **OUTPUT_FILE**

Especifica o caminho do arquivo onde será salvo o arquivamento de log. Se isso for fornecido, os logs serão salvos localmente. Caso contrário, eles serão enviados para o Amazon S3 com o URL retornado na saída. O padrão é fazer upload para o Amazon S3.

--region, -r **REGION**

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--start-time **START_TIME**

Especifica o início do intervalo de tempo, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo 2021-01-01T20:00:00Z). Eventos de log com uma marca de data/hora igual ou posterior a esse horário estão incluídos. Se não for especificado, o padrão é a hora em que o cluster foi criado.

Exemplo usando a AWS ParallelCluster versão 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 os eventos associados a um fluxo de logs.

```
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]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster get-cluster-log-events`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--log-stream-name *LOG_STREAM_NAME*

Especifica o nome do fluxo de logs. Você pode usar o comando `list-cluster-log-streams` para recuperar um fluxo de logs associado a um evento ou eventos.

--debug

Habilitar log de depuração

--end-time *END_TIME*

Especifica o final do intervalo de tempo, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo 2021-01-01T20:00:00Z). Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos.

--limit *LIMIT*

Especifica o número de eventos de log retornados pela consulta. Se um valor não for especificado, o máximo será o máximo de eventos de log que couberem em um tamanho de resposta de 1 MB, com até 10.000 eventos de log.

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--start-from-head *START_FROM_HEAD*

Se o valor for `true`, os primeiros eventos de log serão retornados primeiro. Se o valor for `false`, os eventos de log mais recentes serão retornados primeiro. (Padronizado como `false`.)

--start-time *START_TIME*

Especifica o início do intervalo de tempo, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo `2021-01-01T20:00:00Z`). Eventos com uma marca de data/hora igual ou posterior a esse horário estão incluídos.

Exemplo usando a AWS ParallelCluster versão 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 os eventos associados a uma pilha do cluster especificado.

Note

A partir da versão 3.6.0, AWS ParallelCluster usa pilhas aninhadas para criar os recursos associados às filas e aos recursos computacionais. A API `GetClusterStackEvents` e o comando `pcluster get-cluster-stack-events` retornam apenas os eventos da pilha principal do cluster. Você pode visualizar os eventos da pilha de clusters, incluindo aqueles relacionados a filas e recursos computacionais, no console. CloudFormation

```
pcluster get-cluster-stack-events [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster get-cluster-stack-events`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 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 os eventos associados a uma criação de imagem.

```
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]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster get-image-log-events`.

--image-id, -i *IMAGE_ID*

Especifica o ID da imagem.

--log-stream-name *LOG_STREAM_NAME*

Especifica o nome do fluxo de logs. Você pode usar o comando `list-image-log-streams` para recuperar um fluxo de logs associado a um evento ou eventos.

--debug

Habilitar log de depuração

--end-time *END_TIME*

Especifica o final do intervalo de tempo, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo `2021-01-01T20:00:00Z`). Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos.

--limit *LIMIT*

Especifica o número de eventos de log retornados pela consulta. Se um valor não for especificado, o máximo será o máximo de eventos de log que couberem em um tamanho de resposta de 1 MB, com até 10.000 eventos de log.

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--start-from-head *START_FROM_HEAD*

Se o valor for `true`, os primeiros eventos de log serão retornados primeiro. Se o valor for `false`, os eventos de log mais recentes serão retornados primeiro. (Padronizado como `false`.)

--start-time *START_TIME*

Especifica o início do intervalo de tempo, expresso no formato ISO 8601 (YYYY-MM-DDThh:mm:ssZ, por exemplo `2021-01-01T20:00:00Z`). Eventos com uma marca de data/hora igual ou posterior a esse horário estão incluídos.

Exemplo usando a AWS ParallelCluster versão 3.1.2:

```
$ pcluster get-image-log-events --image-id custom-linux2-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 os eventos associados a uma pilha de uma criação de imagem especificada.

```
pcluster get-image-stack-events [-h]
    --image-id IMAGE_ID
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster get-image-stack-events`.

--image-id, -i *IMAGE_ID*

Especifica a ID da imagem.

--debug

Habilitar log de depuração

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 3.1.2:

```
$ pcluster get-image-stack-events --image-id custom-linux2-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

Retorna uma lista de clusters existentes.

```

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]

```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `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} ...]
```

Especifica a lista de status de cluster a serem filtrados. (Padronizado como `all`.)

```
--debug
```

Habilitar log de depuração

```
--next-token NEXT_TOKEN
```

Especifica o token a ser usado para solicitações paginadas.

```
--query QUERY
```

Especifica a consulta JMESPath a ser executada na saída.

```
--region, -r REGION
```

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 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 a lista de fluxo de logs associados a um cluster.

```
pcluster list-cluster-log-streams [-h]
    --cluster-name CLUSTER_NAME
    [--filters FILTERS [FILTERS ...]]
    [--next-token NEXT_TOKEN] [--debug]
    [--query QUERY]
    [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster list-cluster-log-streams`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--filters *FILTERS* [*FILTERS* ...]

Especifica filtros para o fluxo de logs. Formato: `Name=a,Values=1 Name=b,Values=2,3`.

Filtros compatíveis para:

`private-dns-name`

Especifica a forma abreviada do nome DNS privado da instância (por exemplo, `ip-10-0-0-101`).

`node-type`

Especifica o tipo de nó, o único valor aceito para esse filtro é `HeadNode`.

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 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 a lista de imagens personalizadas existentes.

```
pcluster list-images [-h]  
  --image-status {AVAILABLE,PENDING,FAILED}  
  [--debug]  
  [--next-token NEXT_TOKEN]  
  [--query QUERY]  
  [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster list-images`.

--image-status {AVAILABLE,PENDING,FAILED}

Filtra as imagens retornadas pelo status fornecido.

--debug

Habilitar log de depuração

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 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 a lista de fluxo de logs associados a uma imagem.

```
pcluster list-image-log-streams [-h]
    --image-id IMAGE_ID
    [--next-token NEXT_TOKEN] [--debug]
    [--query QUERY]
    [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster list-image-log-streams`.

--image-id, -i *IMAGE_ID*

Especifica a ID da imagem.

--debug

Habilitar log de depuração

--next-token *NEXT_TOKEN*

Especifica o token a ser usado para solicitações paginadas.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 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

Descreva as AWS ParallelCluster AMIs oficiais.

```
pcluster list-official-images [-h]  
    [--architecture ARCHITECTURE]  
    [--debug]  
    [--os OS]  
    [--query QUERY]
```



```
[--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster list-official-images`.

--architecture *ARCHITECTURE*

Especifica a arquitetura a ser usada para filtrar os resultados. Se esse parâmetro não for especificado, todas as arquiteturas serão retornadas.

--debug

Habilitar log de depuração

--os *OS*

Especifica o sistema operacional a ser usado para filtrar os resultados. Se esse parâmetro não for especificado, todos os sistemas operacionais serão retornados.

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a configuração [Região](#) no arquivo de configuração da imagem, a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo usando a AWS ParallelCluster versão 3.1.2:

```
$ pcluster list-official-images
{
  "images": [
    {
      "amiId": "ami-015cf4eb4e0d6306b2",
      "os": "ubuntu2004",
      "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-x86_64-202202261505
2022-02-26T15-08-34.759Z",
      "version": "3.1.2",
      "architecture": "x86_64"
```

```
  },
  {
    "amiId": "ami-036f23237ce49d25b",
    "os": "ubuntu2204",
    "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-x86_64-202202261505
2022-02-26T15-08-17.558Z",
    "version": "3.1.2",
    "architecture": "x86_64"
  },
  {
    "amiId": "ami-09e5327e694d89ef4",
    "os": "ubuntu2004",
    "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.736Z",
    "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-0bf6d01f398f3737e",
    "os": "centos7",
    "name": "aws-parallelcluster-3.1.2-centos7-hvm-x86_64-202202261505
2022-02-26T15-08-25.001Z",
    "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

Executa um comando ssh com o nome de usuário de cluster e o endereço IP pré-preenchidos. Os argumentos arbitrários são anexados ao final da linha de comando ssh.

```
pcluster ssh [-h]
              --cluster-name CLUSTER_NAME
              [--debug]
              [--dryrun DRYRUN]
              [--region REGION]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster ssh`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster ao qual conectar-se.

--debug

Habilitar log de depuração

--dryrun *DRYRUN*

Quando `true`, imprime a linha de comando que seria executado e sai. (Padronizado como `false`.)

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de `AWS_DEFAULT_REGION` ambiente, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

Exemplo:

```
$ pcluster ssh --cluster-name mycluster -i ~/.ssh/id_rsa
```

Executa um comando ssh com o nome de usuário e o endereço IP do cluster pré-preenchidos:

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

pcluster update-cluster

Atualiza um cluster existente para corresponder às configurações de um arquivo de configuração especificado.

```
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}]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster update-cluster`.

--cluster-configuration, -c *CLUSTER_CONFIGURATION*

Especifica o arquivo de configuração do cluster YAML.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--debug

Habilitar log de depuração

--dryrun *DRYRUN*

Quando `true`, executa a validação sem atualizar o cluster e sem criar recursos. Ele pode ser usado para validar a configuração da imagem e os requisitos de atualização. (Padronizado como `false`.)

--force-update *FORCE_UPDATE*

Quando `true`, força a atualização ignorando os erros de validação da atualização. (Padronizado como `false`.)

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a [Region](#) configuração no arquivo de configuração do cluster, a variável de ambiente `AWS_DEFAULT_REGION`, a `region` configuração na `[default]` seção do `~/.aws/config` arquivo ou o `--region` parâmetro.

--suppress-validators *SUPPRESS_VALIDATORS* [*SUPPRESS_VALIDATORS ...*]

Identifica um ou mais validadores de configuração a serem suprimidos.

Formato: `(ALL|type:[A-Za-z0-9]+)`

--validation-failure-level *{INFO,WARNING,ERROR}*

Especifica o nível de falhas de validação relatado para a atualização.

Exemplo usando a AWS ParallelCluster versão 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": "pc-beta-test",
        "KeyName": "output",
        "EnableWriteAccess": false
      }
    },
    {
      "parameter": "HeadNode.Iam.S3Access",
      "currentValue": {
        "BucketName": "pcluster-east-test-bucket",
        "KeyName": "logs",
        "EnableWriteAccess": true
      }
    }
  ]
}

```

pcluster update-compute-fleet

Atualiza o status da frota de computação do cluster.

```

pcluster update-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    --status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

    [--debug]
    [--query QUERY]
    [--region REGION]

```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para `pcluster update-compute-fleet`.

--cluster-name, -n *CLUSTER_NAME*

Especifica o nome do cluster.

--status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

Especifica o status aplicado à frota de computação do cluster. Os status START_REQUESTED e STOP_REQUESTED correspondem ao agendador do slurm, enquanto os status a ENABLED e DISABLED correspondem ao agendador. AWS Batch

--debug

Habilitar log de depuração

--query *QUERY*

Especifica a consulta JMESPath a ser executada na saída.

--region, -r *REGION*

Especifica o a Região da AWS ser usado. O Região da AWS deve ser especificado usando a variável de AWS_DEFAULT_REGION ambiente, a region configuração na [default] seção do ~/.aws/config arquivo ou o --region parâmetro.

Exemplo usando a AWS ParallelCluster versão 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

Exibe a versão do AWS ParallelCluster.

```
pcluster version [-h] [--debug]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para pcluster version.

--debug

Habilitar log de depuração

Exemplo usando a AWS ParallelCluster versão 3.1.4:

```
$ pcluster version
{
  "version": "3.1.4"
}
```

pcluster3-config-converter

Lê um arquivo de configuração do AWS ParallelCluster versão 2 e grava um arquivo de configuração do AWS ParallelCluster versão 3.

```
pcluster3-config-converter [-h]
                          [-t CLUSTER_TEMPLATE]
                          [-c CONFIG_FILE]
                          [--force-convert]
                          [-o OUTPUT_FILE]
```

Argumentos nomeados

-h, --help

Mostra o texto de ajuda para pcluster3-config-converter.

-t **CLUSTER_TEMPLATE**, --cluster-template **CLUSTER_TEMPLATE**

Especifica a [seção de \[cluster\]](#) do arquivo de configuração a ser convertida. Se não for especificado, o script procurará o parâmetro [cluster-template](#) na [seção de \[global\]](#) ou pesquisará por [cluster default].

-c **CONFIG_FILE**, --config-file **CONFIG_FILE**

Especifica o arquivo de configuração do AWS ParallelCluster versão 2 a ser lido.

--force-convert

Permite a conversão mesmo se uma ou mais configurações não forem suportadas e não forem recomendadas.

-o **OUTPUT_FILE**, --output-file **OUTPUT_FILE**

Especifica o arquivo de configuração do AWS ParallelCluster versão 3 a ser gravado. Se esse parâmetro não for especificado, a configuração será gravada em stdout.

Note

O comando `pcluster3-config-convert` foi adicionado no AWS ParallelCluster versão 3.0.1.

Arquivos de configuração

AWS ParallelCluster usa arquivos YAML 1.1 para parâmetros de configuração.

Tópicos

- [Arquivo de configuração do cluster](#)
- [Arquivos de configuração de imagem de compilação](#)

Arquivo de configuração do cluster

AWS ParallelCluster a versão 3 usa arquivos de configuração separados para controlar a definição da infraestrutura de cluster e a definição de AMIs personalizadas. Todos os arquivos de configuração usam arquivos YAML 1.1. As informações detalhadas de cada um desses arquivos de configuração estão vinculadas abaixo. Para ver alguns exemplos de configurações, consulte https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs.

Esses objetos são usados para a configuração do cluster da AWS ParallelCluster versão 3.

Tópicos

- [Propriedades do arquivo de configuração do cluster](#)
- [Seção Imds](#)
- [Seção Image](#)
- [Seção HeadNode](#)
- [Seção Scheduling](#)
- [Seção SharedStorage](#)
- [Seção Iam](#)
- [Seção LoginNodes](#)
- [Seção Monitoring](#)
- [Seção Tags](#)

- [Seção AdditionalPackages](#)
- [Seção DirectoryService](#)
- [Seção DeploymentSettings](#)

Propriedades do arquivo de configuração do cluster

Region (Opcional, String)

Especifica o Região da AWS para o cluster. Por exemplo, us-east-2.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

CustomS3Bucket (Opcional, String)

Especifica o nome de um bucket do Amazon S3 criado em Conta da AWS você para armazenar recursos que são usados pelos seus clusters, como o arquivo de configuração do cluster. AWS ParallelCluster mantém um bucket do Amazon S3 em cada um em Região da AWS que você cria clusters. Por padrão, esses buckets do Amazon S3 são denominados `parallelcluster-hash-v1-D0-NOT-DELETE`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida. Se você forçar a atualização, o novo valor será ignorado e o valor antigo será usado.](#)

AdditionalResources (Opcional, String)

Define um AWS CloudFormation modelo adicional a ser executado junto com o cluster. Esse modelo adicional é usado para a criação de recursos que existem fora do cluster, mas que fazem parte do ciclo de vida do cluster.

Esse valor deve ser um URL HTTPS para um modelo público, com todos os parâmetros fornecidos.

Não há valor padrão.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Seção **Imds**

(Opcional) Especifica a configuração do serviço de metadados da instância global (IMDS).

[Imds :](#)

`ImdsSupport: string`

Propriedades do **Imds**

ImdsSupport (Opcional, String)

Especifica quais versões do IMDS são suportadas nos nós do cluster. Os valores compatíveis são `v1.0` e `v2.0`. O valor padrão é `v2.0`.

Se `ImdsSupport` estiver definido como `v1.0`, há compatibilidade com o IMDSv1 e o IMDSv2.

Se `ImdsSupport` estiver definido como `v2.0`, somente o IMDSv2 será suportado.

Para ter mais informações, consulte [Usar o IMDSv2](#), no Guia do usuário do EC2 para instâncias Linux.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

A partir da AWS ParallelCluster versão 3.7.0, o valor `ImdsSupport` padrão é `v2.0`. Recomendamos que você configure `ImdsSupport` para `v2.0` e substitua o IMDSv1 pelo IMDSv2 em suas chamadas de ações personalizadas.

O suporte para [Imds](#) foi [ImdsSupport](#) adicionado com a AWS ParallelCluster versão 3.3.0.

Seção **Image**

(Obrigatório) Define o sistema operacional do cluster.

`Image:`

`Os: string`

`CustomAmi: string`

Propriedades do **Image**

Os (Obrigatório, String)

Especifica o sistema operacional a ser usado para o cluster. Os valores suportados são `linux2`, `centos7`, `ubuntu2204`, `ubuntu2004`, `rhel8`, `rocky8`, `rhel9`, `rocky9`.

Note

RedHat O Enterprise Linux 8.7 (rhel8) é adicionado a partir da AWS ParallelCluster versão 3.6.0.

Se você configurar seu cluster para usar `rhel`, o custo sob demanda para qualquer tipo de instância será maior do que quando você configura seu cluster para usar outros sistemas operacionais compatíveis. Para obter mais informações sobre preços, consulte [Preços sob demanda](#) e [Como o Red Hat Enterprise Linux no Amazon EC2 é oferecido e definido seu preço?](#).

RedHat O Enterprise Linux 9 (rhel9) é adicionado a partir da AWS ParallelCluster versão 3.9.0.

Além dos específicos Regiões da AWS mencionados na tabela a seguir, que não são compatíveis com `centos7`. Todas as outras regiões AWS comerciais oferecem suporte a todos os sistemas operacionais a seguir.

Partição (Regiões da AWS)	<code>alinux2</code>	<code>centos7</code>	<code>ubuntu2204</code> e <code>ubuntu2004</code>	<code>rhel8</code>	<code>rhel9</code>
Comercial (tudo Regiões da AWS não mencionadas especificamente)	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro
AWS GovCloud (Leste dos EUA) (<code>us-gov-east-1</code>)	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro
AWS GovCloud (Oeste dos EUA) (<code>us-gov-west-1</code>)	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro
China (Pequim) (<code>cn-north-1</code>)	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro

Partição (Regiões da AWS)	alinux2	centos7	ubuntu2204 e ubuntu2004	rhel8	rhel9
China (Ningxia) (cn-northwest-1)	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

AWS ParallelCluster O 3.8.0 oferece suporte ao Rocky Linux 8, mas as AMIs pré-construídas do Rocky Linux 8 (para arquiteturas x86 e ARM) não estão disponíveis. AWS ParallelCluster A versão 3.8.0 suporta a criação de clusters com o Rocky Linux 8 usando AMIs personalizadas. Para obter mais informações, consulte [Considerações sobre o sistema operacional](#). AWS ParallelCluster O 3.9.0 oferece suporte ao Rocky Linux 9, mas as AMIs pré-construídas do Rocky Linux 9 (para arquiteturas x86 e ARM) não estão disponíveis. AWS ParallelCluster A versão 3.9.0 suporta a criação de clusters com o Rocky Linux 9 usando AMIs personalizadas. Para obter mais informações, consulte [Considerações sobre o sistema operacional](#).

CustomAmi (Opcional, String)

Especifica o ID de uma AMI personalizada a ser usada para os nós principal e de computação em vez da AMI padrão. Para ter mais informações, consulte [AWS ParallelCluster Personalização da AMI](#).

Se a AMI personalizada exigir permissões adicionais para seu lançamento, essas permissões deverão ser adicionadas às políticas do usuário e do nó principal.

Por exemplo, se uma AMI personalizada tiver um snapshot criptografado associado a ela, as seguintes políticas adicionais serão necessárias nas políticas do usuário e do nó principal:

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

```

    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
      ]
    }
  ]
}

```

Para criar uma AMI personalizada do RedHat Enterprise Linux, você deve configurar o sistema operacional para instalar os pacotes fornecidos pelos repositórios RHUI (AWS): `rhel-<version>-baseos-rhui-rpms`, e `rhel-<version>-appstream-rhui-rpms` e `codeready-builder-for-rhel-<version>-rhui-rpms`. Além disso, os repositórios na AMI personalizada devem conter pacotes `kernel-devel` na mesma versão da versão do kernel em execução.

Limitações conhecidas:

- Somente o RHEL 8.2 e versões posteriores oferecem suporte ao FSx para Lustre.
- A versão 4.18.0-425.3.1.el8 do kernel do RHEL 8.7 não oferece suporte a FSx para Lustre.
- Somente o RHEL 8.4 e versões posteriores oferecem suporte a EFA.

Para solucionar problemas de avisos de validação da AMI personalizada, consulte [Solução de problemas de AMI personalizada](#).

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Seção HeadNode

(Obrigatório) Especifica a configuração do nó principal.

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*

Propriedades do **HeadNode**

InstanceType (Obrigatório, String)

Especifica o tipo de instância do nó principal.

Especifica o tipo de instância do Amazon EC2 que é usado para o nó principal. A arquitetura do tipo de instância deve ser a mesma usada para a Slurm [InstanceType](#) configuração AWS Batch [InstanceType](#) ou.

Note

AWS ParallelCluster não é compatível com os seguintes tipos de instância para a HeadNode configuração.

- hpc6id

Se você definir um tipo de instância p4d ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede, você deve definir [ElasticIp](#) para `true` para fornecer acesso público. IPs públicos da AWS só podem ser atribuídos a instâncias iniciadas

com uma única interface de rede. Nesse caso, recomendamos que você use um [gateway NAT](#) para fornecer acesso público aos nós de computação do cluster. Para obter mais informações, consulte [Atribuir um endereço IPv6 a uma instância](#), no Guia do usuário do Amazon EC2 para instâncias Linux.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

`DisableSimultaneousMultithreading` (Opcional, Boolean)

Set true, desabilita o hyperthreading no nó principal. O valor padrão é false.

Nem todos os tipos de instância podem desabilitar o hyperthreading. Para obter uma lista de tipos de instância que suportam a desativação do hyperthreading, consulte [Núcleos e threads de CPU para cada núcleo de CPU para cada tipo de instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

`SharedStorageType` (Opcional, String)

Especifica o tipo de armazenamento usado para dados compartilhados internamente. Os dados compartilhados internamente incluem dados AWS ParallelCluster usados para gerenciar o cluster e os compartilhados padrão, `/home` se não forem especificados no diretório [Seção SharedStorage](#) as a `Mount`, para montar um volume compartilhado do sistema de arquivos. Para obter mais detalhes sobre dados compartilhados internos, consulte [AWS ParallelCluster Diretórios internos](#).

Se `Ebs`, que é o tipo de armazenamento padrão, o nó principal exportará partes de seu volume raiz como diretórios compartilhados para nós de computação e nós de login usando NFS.

Se `Efs`, o Parallelcluster criará um sistema de arquivos EFS para usar para dados internos compartilhados e `/home`

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

Quando o cluster se expande, o tipo de armazenamento do EBS pode apresentar gargalos de desempenho, pois o nó principal compartilha dados do volume raiz com os nós de computação usando exportações NFS. Usando o EFS, você pode evitar exportações de NFS à medida que seu cluster se expande e evitar gargalos de desempenho associados a elas. É recomendável escolher o EBS para obter o máximo

potencial de leitura/gravação para arquivos pequenos e para o processo de instalação.
Escolha EFS para escalar.

Networking

(Obrigatório) Define a configuração de rede para o nó principal.

Networking:

SubnetId: *string*

ElasticIp: *string/boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

Proxy:

HttpProxyAddress: *string*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do **Networking**

SubnetId (Obrigatório, String)

Especifica o ID de uma sub-rede existente na qual o servidor provisiona o nó principal.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

ElasticIp (Opcional, String)

Cria ou atribui um endereço IP elástico ao nó principal. Os valores suportados são true, false ou o ID de um endereço IP elástico existente. O padrão é false.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

SecurityGroups (Opcional, [String])

Lista de IDs de grupos de segurança do Amazon VPC a serem usados no nó principal. Eles substituem os grupos de segurança AWS ParallelCluster criados se essa propriedade não for incluída.

Verifique se os grupos de segurança estão configurados corretamente para seus [SharedStorage](#) sistemas.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

`AdditionalSecurityGroups` (Opcional, `[String]`)

Lista de IDs adicionais de grupos de segurança do Amazon VPC para usar no nó principal.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

`Proxy` (Opcional)

Especifica as configurações de proxy para o nó principal.

`Proxy:`

`HttpProxyAddress`: *string*

`HttpProxyAddress` (Opcional, `String`)

Define um servidor de proxy HTTP ou HTTPS, normalmente `https://x.x.x.x:8080`.

Não há valor padrão.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Ssh

(Opcional) Define a configuração para acesso SSH ao nó principal.

`Ssh:`

`KeyName`: *string*

`AllowedIps`: *string*

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Propriedades do **Ssh**

`KeyName` (Opcional, `String`)

Nomeia um par de chaves existente do Amazon EC2 para habilitar o acesso SSH ao nó principal.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

`AllowedIps` (Opcional, `String`)

Especifica o intervalo IP formatado em CIDR ou um ID de lista de prefixos para conexões SSH com o nó principal. O padrão é `0.0.0.0/0`.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

LocalStorage

(Opcional) Define a configuração de armazenamento local para o nó principal.

```
LocalStorage:  
  RootVolume:  
    Size: integer  
    Encrypted: boolean  
    VolumeType: string  
    Iops: integer  
    Throughput: integer  
    DeleteOnTermination: boolean  
  EphemeralVolume:  
    MountDir: string
```

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Propriedades do LocalStorage

RootVolume (Obrigatório)

Especifica o armazenamento do volume raiz para o nó principal.

```
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  VolumeType: string  
  Iops: integer  
  Throughput: integer  
  DeleteOnTermination: boolean
```

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Size (Opcional, Integer)

Especifica o tamanho do volume raiz do nó principal em gibibytes (GiB). O tamanho padrão vem da AMI. Usar um tamanho diferente exige que AMI seja compatível com growroot.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Encrypted (Opcional, Boolean)

Especifica se o volume raiz está criptografado. O valor padrão é `true`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

VolumeType (Opcional, String)

Especifica o [tipo de volume do Amazon EBS](#). Os valores compatíveis são `gp2`, `gp3`, `io1`, `io2`, `sc1`, `st1` e `standard`. O valor padrão é `gp3`.

Para ter mais informações, consulte [Tipos de volumes do Amazon EBS](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Iops (Opcional, Integer)

Define o número de IOPS para os volumes do tipo `io1`, `io2` e `gp3`.

O valor padrão, os valores suportados e a proporção do `volume_iops` para `volume_size` variam de acordo com `VolumeType` e `Size`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

VolumeType = `io1`

Padrão Iops = 100

Valores suportados Iops = 100–64000 †

Proporção máxima de Iops para Size = 50 IOPS por GiB. 5000 IOPS exigem um Size de pelo menos 100 GiB.

VolumeType = `io2`

Padrão Iops = 100

Valores suportados Iops = 100–64000 (256000 para volumes do `io2 Block Express`) †

Proporção máxima de Iops para Size = 500 IOPS por GiB. 5000 IOPS exigem um Size de pelo menos 10 GiB.

VolumeType = `gp3`

Padrão Iops = 3000

Valores suportados Iops = 3000–16000

Proporção máxima de Iops para Size = 500 IOPS por GiB. 5000 IOPS exigem um Size de pelo menos 10 GiB.

† O número máximo de IOPS é garantido somente em [instâncias criadas no Nitro System](#) provisionadas com mais de 32.000 IOPS. Outras instâncias garantem até 32.000 IOPS. Os volumes de io1 mais antigos podem não atingir o desempenho total, a menos que você [modifique o volume](#). io2 Os volumes do Block Express oferecem suporte a valores Iops de até 256000 em tipos de instância R5b. Para obter mais informações, consulte [Volumes do io2 Block Express](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Throughput (Opcional, Integer)

Define o throughput para tipos de volume gp3, em MiB/s. Isso é válido somente quando VolumeType é gp3. O valor padrão é 125. Valores suportados: 125–1000 MiB/s

A proporção de Throughput para Iops não pode ser superior a 0,25. O throughput máximo de 1000 MiB/s exige que a configuração Iops seja de pelo menos 4000.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

DeleteOnTermination (Opcional, Boolean)

Especifica se o volume raiz deve ser excluído quando o nó principal é encerrado. O valor padrão é true.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

EphemeralVolume (Opcional)

Especifica os detalhes de qualquer volume de armazenamento de instâncias. Para obter mais informações, consulte [Volumes de armazenamento de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

[EphemeralVolume:](#)
[MountDir:](#) *string*

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

MountDir (Opcional, String)

Especifica o diretório de montagem para o volume de armazenamento da instância. O padrão é /scratch.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Dcv

(Opcional) Define as definições de configuração para o servidor do NICE DCV em execução no nó do cabeçalho.

Para ter mais informações, consulte [Conecte-se ao nó principal por meio do NICE DCV](#).

Dcv:

`Enabled`: *boolean*
`Port`: *integer*
`AllowedIps`: *string*

Important

Por padrão, a porta NICE DCV configurada por AWS ParallelCluster está aberta a todos os endereços IPv4. No entanto, você pode se conectar a uma porta do NICE DCV somente se tiver o URL da sessão do NICE DCV e se conectarem à sessão do NICE DCV dentro de 30 segundos após o URL ser retornado de `pc1uster dcv-connect`. Use a configuração `AllowedIps` para restringir ainda mais o acesso à porta do NICE DCV com um intervalo IP no formato CIDR e use a configuração `Port` para definir uma porta não padrão.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do Dcv

Enabled (Obrigatório, Boolean)

Especifica se o NICE DCV está habilitado no nó principal. O valor padrão é `false`.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

O NICE DCV gera automaticamente um certificado autoassinado que é usado para proteger o tráfego entre o cliente do NICE DCV e o servidor do NICE DCV em execução

no nó do cabeçalho. Para configurar seu próprio certificado, consulte [Certificado HTTPS do NICE DCV](#).

Port (Opcional, Integer)

Especifica a porta para o NICE DCV. O valor padrão é 8443.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

AllowedIps (Opcional, Recomendado, String)

Especifica o intervalo de IP formatado em CIDR para conexões com o NICE DCV. Essa configuração é usada somente ao AWS ParallelCluster criar o grupo de segurança. O valor padrão é `0.0.0.0/0` e permite o acesso a partir de qualquer endereço de Internet.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

CustomActions

(Opcional) Especifica scripts personalizados para execução no nó principal.

```
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
```

Propriedades do **CustomActions**

OnNodeStart (Opcional)

Especifica um único script ou uma sequência de scripts a serem executados no nó principal antes que qualquer ação de bootstrap de implantação do nó seja iniciada. Para ter mais informações, consulte [Ações de bootstrap personalizadas](#).

Sequence (Opcional)

Lista de scripts a serem executados. AWS ParallelCluster executa os scripts na mesma ordem em que estão listados no arquivo de configuração, começando com o primeiro.

Script (Obrigatório, String)

Especifica o arquivo a ser usado. O caminho do arquivo pode começar com `https://` ou `s3://`.

Args (Opcional, [String])

Lista de argumentos a serem transmitidos para o script.

Script (Obrigatório, String)

Especifica o arquivo a ser usado para um único script. O caminho do arquivo pode começar com `https://` ou `s3://`.

Args (Opcional, [String])

Lista de argumentos a serem transmitidos para o script.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

OnNodeConfigured (Opcional)

Especifica um único script ou uma sequência de scripts a serem executados no nó principal após a conclusão das ações de bootstrap do nó. Para ter mais informações, consulte [Ações de bootstrap personalizadas](#).

Sequence (Opcional)

Especifica a lista de scripts a serem executados.

Script (Obrigatório, String)

Especifica o arquivo a ser usado. O caminho do arquivo pode começar com `https://` ou `s3://`.

Args (Opcional, [String])

Lista de argumentos a serem transmitidos para o script.

Script (Obrigatório, String)

Especifica o arquivo a ser usado para um único script. O caminho do arquivo pode começar com `https://` ou `s3://`.

Args (Opcional, [String])

Lista de argumentos a serem transmitidos para o script.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

OnNodeUpdated (Opcional)

Especifica um único script ou uma sequência de scripts a serem executados no nó principal após a conclusão das ações de atualização do nó. Para ter mais informações, consulte [Ações de bootstrap personalizadas](#).

Sequence (Opcional)

Especifica a lista de scripts a serem executados.

Script (Obrigatório, String)

Especifica o arquivo a ser usado. O caminho do arquivo pode começar com `https://` ou `s3://`.

Args (Opcional, [String])

Lista de argumentos a serem transmitidos para o script.

Script (Obrigatório, String)

Especifica o arquivo a ser usado para um único script. O caminho do arquivo pode começar com `https://` ou `s3://`.

Args (Opcional, [String])

Lista de argumentos a serem transmitidos para o script.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

OnNodeUpdated é adicionado a partir de AWS ParallelCluster 3.4.0.

Sequence é adicionado a partir da AWS ParallelCluster versão 3.6.0. Ao especificar Sequence, você pode listar vários scripts para uma ação personalizada. AWS ParallelCluster continua oferecendo suporte à configuração de uma ação personalizada com um único script, sem incluir Sequence.

AWS ParallelCluster não suporta a inclusão de um único script e Sequence da mesma ação personalizada.

Iam

(Opcional) Especifica uma função de instância ou um perfil de instância a ser usado no nó principal para substituir a função de instância padrão ou o perfil de instância do cluster.

Iam:

InstanceRole: *string*

InstanceProfile: *string*

S3Access:

- BucketName: *string*

EnableWriteAccess: *boolean*

KeyName: *string*

AdditionalIamPolicies:

- Policy: *string*

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Propriedades do Iam

InstanceProfile (Opcional, String)

Especifica um perfil de instância para substituir o perfil padrão da instância do nó principal.

Não é possível especificar ambos InstanceProfile e InstanceRole. O formato é `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

Se isso for especificado, as configurações `S3Access` e `AdditionalIamPolicies` não poderão ser especificadas.

Recomendamos que você especifique uma ou ambas `S3Access` as `AdditionalIamPolicies` configurações, pois os recursos adicionados AWS ParallelCluster geralmente exigem novas permissões.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

InstanceRole (Opcional, String)

Especifica uma função de instância para substituir o a função padrão da instância do nó principal. Não é possível especificar ambos `InstanceProfile` e `InstanceRole`. O formato é `arn:Partition:iam::Account:role/RoleName`.

Se isso for especificado, as configurações `S3Access` e `AdditionalIamPolicies` não poderão ser especificadas.

Recomendamos que você especifique uma ou ambas `S3Access` as `AdditionalIamPolicies` configurações, pois os recursos adicionados AWS ParallelCluster geralmente exigem novas permissões.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

S3Access

S3Access (Opcional)

Especifica um bucket. Isso é usado para gerar políticas para conceder o acesso especificado ao bucket.

Se isso for especificado, as configurações `InstanceProfile` e `InstanceRole` não poderão ser especificadas.

Recomendamos que você especifique uma ou ambas `S3Access` as `AdditionalIamPolicies` configurações, pois os recursos adicionados AWS ParallelCluster geralmente exigem novas permissões.

S3Access:

- `BucketName`: *string*

`EnableWriteAccess`: *boolean*

`KeyName`: *string*

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

BucketName (Obrigatório, String)

O nome do bucket do .

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

KeyName (Opcional, String)

A chave para o bucket. O valor padrão é "*".

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

EnableWriteAccess (Opcional, Boolean)

Indica se o acesso de gravação está habilitado para o bucket. O valor padrão é false.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

AdditionalIamPolicies

AdditionalIamPolicies (Opcional)

Especifica uma lista de nomes de recursos da Amazon (ARNs) de políticas do IAM para o Amazon EC2. Essa lista é anexada à função raiz usada para o nó principal, além das permissões exigidas pelo AWS ParallelCluster.

O nome da política do IAM e seu ARN são diferentes. Os nomes não podem ser usados.

Se isso for especificado, as configurações InstanceProfile e InstanceRole não poderão ser especificadas.

Recomendamos que você use AdditionalIamPolicies porque AdditionalIamPolicies são adicionados às permissões necessárias e InstanceRole devem incluir todas as permissões necessárias. AWS ParallelCluster As permissões necessárias muitas vezes são alteradas entre as versões à medida que recursos são adicionados.

Não há valor padrão.

AdditionalIamPolicies:

- **Policy:** *string*

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Policy (Opcional, [String])

Lista de políticas do IAM.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Imds

(Opcional) Especifica as propriedades do serviço de metadados da instância (IMDS). Para obter mais informações, consulte [Como funciona o serviço de metadados de instância versão 2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Imds:

Secured: *boolean*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do Imds

Secured (Opcional, Boolean)

Se `true`, restringe o acesso ao IMDS do nó principal (e às credenciais do perfil da instância) a um subconjunto de superusuários.

Se `false`, cada usuário no nó principal tem acesso ao IMDS do nó principal.

Os seguintes usuários têm permissão para acessar o IMDS do nó principal:

- usuário raiz
- usuário administrativo do cluster (`pc-cluster-admin` por padrão)
- usuário padrão específico do sistema operacional (`ec2-user` no Amazon Linux 2 e RedHat, `ubuntu` no Ubuntu 18.04, `centos` no CentOS 7)

O padrão é `true`.

Os default usuários são responsáveis por garantir que um cluster tenha as permissões necessárias para interagir com AWS os recursos. Se você desabilitar default o acesso do usuário ao IMDS, não AWS ParallelCluster conseguirá gerenciar os nós de computação e deixará de funcionar. Não desative o acesso do usuário default ao IMDS.

Quando um usuário recebe acesso ao IMDS do nó principal, ele pode usar as permissões incluídas no [perfil da instância do nó principal](#). Por exemplo, eles podem usar essas permissões para iniciar instâncias do EC2 ou ler a senha de um domínio do AD que o cluster está configurado para usar para autenticação.

Para restringir o acesso ao IMDS, AWS ParallelCluster gerencia uma cadeia de iptables.

Os usuários do cluster com acesso sudo podem ativar ou desativar seletivamente o acesso ao IMDS do nó principal para outros usuários individuais, incluindo usuários default, executando o comando:

```
$ sudo /opt/parallelcluster/scripts/imds/imds-access.sh --allow <USERNAME>
```

Você pode desativar o acesso do usuário ao IMDS com a opção --deny desse comando.

Se você desabilitar inadvertidamente o acesso do usuário default ao IMDS, poderá restaurar a permissão usando a opção --allow.

Note

Qualquer personalização de regras iptables ou ip6tables pode interferir no mecanismo usado para restringir o acesso ao IMDS no nó principal.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Image

(Opcional) Define uma imagem personalizada para o nó principal.

Image:

CustomAmi: *string*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do **Image**

CustomAmi (Opcional, String)

Especifica o ID de uma AMI personalizada a ser usada para o nó principal em vez da AMI padrão. Para ter mais informações, consulte [AWS ParallelCluster Personalização da AMI](#).

Se a AMI personalizada exigir permissões adicionais para seu lançamento, essas permissões deverão ser adicionadas às políticas do usuário e do nó principal.

Por exemplo, se uma AMI personalizada tiver um snapshot criptografado associado a ela, as seguintes políticas adicionais serão necessárias nas políticas do usuário e do nó principal:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>;key/<AWS_KMS_KEY_ID>"
      ]
    }
  ]
}
```

Para solucionar problemas de avisos de validação da AMI personalizada, consulte [Solução de problemas de AMI personalizada](#).

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Seção **Scheduling**

(Obrigatório) Define o programador de trabalhos usado no cluster e as instâncias de computação que o programador de trabalhos gerencia. Você pode usar o AWS Batch agendador Slurm ou. Cada um oferece suporte a um conjunto diferente de configurações e propriedades.

Tópicos

- [Propriedades do 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`

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
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
```

Propriedades do **Scheduling**

Scheduler (Obrigatório, **String**)

Especifica o tipo de programador usado. Os valores compatíveis são `slurm` e `awsbatch`.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

`awsbatch` suporta apenas o sistema operacional `alinux2` e a plataforma `x86_64`.

ScalingStrategy (Opcional, **String**)

Permite que você escolha como os nós dinâmicos do Slurm se expandem. Os valores suportados são `all-or-nothing`, `greedy-all-or-nothing` e `best-effort` o valor padrão é `all-or-nothing`.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

A estratégia de escalabilidade se aplica somente aos nós a serem retomados pelo Slurm, não aos nós que eventualmente já estão em execução.

- **all-or-nothing** Essa estratégia segue estritamente uma all-or-nothing-approach, com o objetivo de evitar instâncias ociosas no final do processo de escalabilidade. Ele opera em uma all-or-nothing base, o que significa que ele se expande completamente ou não cresce. Esteja ciente de que pode haver custos adicionais devido a instâncias lançadas temporariamente, quando os trabalhos exigem mais de 500 nós ou abrangem vários recursos computacionais. Essa estratégia tem o menor rendimento entre as três estratégias de escalonamento possíveis. O tempo de escalonamento depende do número de trabalhos enviados por execução do programa de currículo do Slurm. Além disso, você não pode escalar muito além do limite padrão da conta de RunInstances recursos por execução, que é de 1000 instâncias por padrão. Mais detalhes podem ser encontrados na documentação de [limitação da API AWS EC2](#)
- **greedy-all-or-nothing** Semelhante à all-or-nothing estratégia, ela visa evitar instâncias ociosas após o escalonamento. Essa estratégia permite a superescalabilidade temporária durante o processo de escalabilidade para obter uma taxa de transferência maior do que a all-or-nothing abordagem, mas também vem com o mesmo limite de escalabilidade de 1.000 instâncias de acordo com o limite da conta de recursos. RunInstances
- **best-effort** Essa estratégia prioriza a alta taxa de transferência, mesmo que isso signifique que algumas instâncias possam ficar inativas no final do processo de escalabilidade. Ele tenta alocar quantos nós forem solicitados pelos trabalhos, mas existe a possibilidade de não atender a solicitação inteira. Diferentemente das outras estratégias, a abordagem de melhor esforço pode acumular mais instâncias do que o RunInstances limite padrão, ao custo de ter recursos ociosos ao longo das várias execuções do processo de escalabilidade.

Cada estratégia é projetada para atender às diferentes necessidades de escalabilidade, permitindo que você selecione uma que atenda aos seus requisitos e restrições específicos.

AwsBatchQueues

(Opcional) As configurações da AWS Batch fila. Somente uma fila é suportada. Se [Scheduler](#) estiver definido como `awsbatch`, essa seção é obrigatória. Para obter mais informações sobre o programador `awsbatch`, consulte [configuração de redes](#) e [AWS Batch \(awsbatch\)](#).

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
```

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Propriedades do **AwsBatchQueues**

Name (Obrigatório, **String**)

O nome da AWS Batch fila.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

CapacityType (Opcional, **String**)

O tipo de recursos computacionais que a AWS Batch fila usa. Os valores suportados são ONDEMAND, SPOT ou CAPACITY_BLOCK. O valor padrão é ONDEMAND.

Note

Se você definir CapacityType como SPOT, sua conta deverá conter uma função vinculada ao serviço AWSServiceRoleForEC2Spot. Você pode criar essa função usando o AWS CLI comando a seguir.

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

Para obter mais informações, consulte [Função vinculada ao serviço para solicitações de instâncias spot](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

Networking

(Obrigatório) Define a configuração de rede para a AWS Batch fila.

```
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
```

Propriedades do **Networking**

SubnetIds (Obrigatório, **[String]**)

Especifica o ID de uma sub-rede existente para provisionar a AWS Batch fila. No momento, somente uma sub-rede é compatível.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

AssignPublicIp (Opcional, **String**)

Cria ou atribui um endereço IP público aos nós na AWS Batch fila. Os valores compatíveis são `true` e `false`. O padrão depende da sub-rede que você especificou.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

SecurityGroups (Opcional, **[String]**)

Lista de grupos de segurança que a AWS Batch fila usa. Se você não especificar grupos de segurança, AWS ParallelCluster cria novos grupos de segurança.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

AdditionalSecurityGroups (Opcional, **[String]**)

Lista de grupos de segurança que a AWS Batch fila usa.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

ComputeResources

(Obrigatório) Define a ComputeResources configuração da AWS Batch fila.

```
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
```

Propriedades do **ComputeResources**

Name (Obrigatório, **String**)

O nome do ambiente de computação da AWS Batch fila.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

InstanceTypes (Obrigatório, [**String**])

A matriz de tipos AWS Batch de instância do ambiente computacional. Todos os tipos de instância devem usar a arquitetura x86_64.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

MinvCpus (Opcional, **Integer**)

O número mínimo de vCPUs que um ambiente AWS Batch computacional pode usar.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

DesiredVcpus (Opcional, **Integer**)

O número desejado de vCPUs no ambiente AWS Batch computacional. AWS Batch ajusta esse valor entre MinvCpus e MaxvCpus com base na demanda na fila de trabalhos.

Política de atualização: essa configuração não é analisada durante uma atualização.

MaxvCpus (Opcional, **Integer**)

O número máximo de vCPUs para o ambiente AWS Batch computacional. Você não pode definir isso para um valor menor que o DesiredVcpus.

Política de atualização: essa configuração não pode ser diminuída durante uma atualização.

SpotBidPercentage (Opcional, **Float**)

A porcentagem máxima que o preço On-Demand para esse tipo de instância que uma instância spot do EC2 pode atingir antes que as instâncias sejam iniciadas. O valor padrão é 100 (100%). O intervalo compatível é 1-100.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

SlurmQueues

(Opcional) Configurações para a fila Slurm. Se [Scheduler](#) estiver definido como slurm, essa seção é obrigatória.

```
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
  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
```

Política de atualização: para essa configuração de valores de lista, um novo valor pode ser adicionado durante uma atualização ou a frota de computação deve ser interrompida ao remover um valor existente.

Propriedades do **SlurmQueues**

Name (Obrigatório, **String**)

O nome da fila do Slurm.

Note

O tamanho do cluster pode mudar durante uma atualização. Para obter mais informações, consulte [Capacidade, tamanho e atualização do cluster](#)

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

CapacityReservationTarget**Note**

CapacityReservationTarget é adicionado com a AWS ParallelCluster versão 3.3.0.

CapacityReservationTarget:

CapacityReservationId: *string*

CapacityReservationResourceGroupArn: *string*

Especifica a reserva de capacidade sob demanda para os recursos de computação da fila.

CapacityReservationId (Opcional, **String**)

O ID da reserva de capacidade existente a ser direcionada para os recursos de computação da fila. O id pode se referir a um [ODCR](#) ou a um [bloco de capacidade para ML](#).

A reserva deve usar a mesma plataforma usada pela instância. Por exemplo, se suas instâncias são executadas em `rhe18`, sua reserva de capacidade deve ser executada na plataforma Red Hat Enterprise Linux. Para obter mais informações, consulte [Plataformas compatíveis](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Note

Se você incluir [Instances](#) na configuração do cluster, deverá excluir essa configuração de nível de fila da configuração `CapacityReservationId`.

CapacityReservationResourceGroupArn (Opcional, **String**)

O nome do recurso da Amazon (ARN) do grupo de recursos que serve como o grupo de reservas de capacidade vinculadas ao serviço para os recursos de computação da fila. AWS

ParallelCluster identifica e usa a reserva de capacidade mais adequada do grupo de recursos com base nas seguintes condições:


- Se PlacementGroup estiver habilitado em [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleciona um grupo de recursos que tem como alvo o tipo de instância e PlacementGroup para um recurso computacional, se o recurso computacional existir.

PlacementGroup precisa ter como alvo um dos tipos de instância definidos em [ComputeResources](#).

- Se PlacementGroup não estiver habilitado em [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleciona um grupo de recursos que tem como alvo somente o tipo de instância de um recurso computacional, se o recurso computacional existir.

O grupo de recursos deve ter pelo menos um ODCR para cada tipo de instância reservado em uma zona de disponibilidade em todos os recursos de computação e zonas de disponibilidade da fila. Para ter mais informações, consulte [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#).

Para obter mais informações sobre os requisitos de configuração de várias sub-redes, consulte [Networking](#) / [SubnetIds](#).

 Note

Várias zonas de disponibilidade foram adicionadas na AWS ParallelCluster versão 3.4.0.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

CapacityType (Opcional, **String**)

Os tipos de recursos de computação que a fila Slurm usa. Os valores compatíveis são ONDEMAND ou SPOT. O valor padrão é ONDEMAND.

Note

Se você definir `CapacityType` para `SPOT`, sua conta deverá conter uma função vinculada ao serviço `AWSServiceRoleForEC2Spot`. Você pode criar essa função usando o AWS CLI comando a seguir.

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

Para obter mais informações, consulte [Função vinculada ao serviço para solicitações de instâncias spot](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Política de atualização: a frota de computação deve ser interrompida ou `QueueUpdateStrategy` para que essa configuração seja alterada para uma atualização.

AllocationStrategy (Opcional, **String**)

Especifique a estratégia de alocação para todos os recursos de computação definidos em [Instances](#).

Valores válidos: `lowest-price` | `capacity-optimized`

Padrão: `lowest-price`

lowest-price


- Se você especificar `CapacityType = ONDEMAND`, a Frota do EC2 usará o preço para determinar o pedido, executando primeiro as instâncias de preços mais baixos.
- Se você usar `CapacityType = SPOT`, a Frota do EC2 executa instâncias do grupo de instâncias spot de menor preço que possuam capacidade disponível. Se um grupo ficar sem capacidade antes de cumprir sua capacidade exigida, a Frota do EC2 atenderá à sua solicitação lançando instâncias para você. Particularmente, a Frota do EC2 executa instâncias do grupo de instâncias spot de menor preço que possuam capacidade disponível. A Frota do EC2 pode lançar instâncias spot de vários grupos diferentes.
- Se você definir `CapacityType = CAPACITY_BLOCK`, não há estratégias de alocação, portanto, o `AllocationStrategy` parâmetro não pode ser configurado.

capacity-optimized

- Se você definir `CapacityType = ONDEMAND`, `capacity-optimized` não ficará disponível.

- Se você definir `CapacityType = SPOT`, a Frota do EC2 executará instâncias de grupos de instâncias spot com capacidade ideal para o número de instâncias que serão executadas.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

 Note

`AllocationStrategy` é suportado a partir do AWS ParallelCluster versão 3.3.0.


JobExclusiveAllocation (Opcional, **String**)

Se definido como `true`, o sinalizador `OverSubscribe` de partição Slurm será definido como `EXCLUSIVE`. Quando `OverSubscribe=EXCLUSIVE`, os trabalhos na partição têm acesso exclusivo a todos os nós alocados. Para obter mais informações, consulte [EXCLUSIVE](#) na documentação do Slurm.

Valores válidos: `true` | `false`

Padrão: `false`

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

 Note

`JobExclusiveAllocation` é suportado a partir do AWS ParallelCluster versão 3.7.0.

CustomSlurmSettings (Opcional, **Dict**)

Define as configurações personalizadas da partição (fila) Slurm.

Especifica um dicionário de pares de valores-chave de parâmetros de configuração personalizados Slurm que se aplicam às filas (partições).

Cada par de valores-chave separado, como `Param1: Value1`, é adicionado separadamente ao final da linha de configuração da partição Slurm no formato `Param1=Value1`.

Você só pode especificar parâmetros de configuração Slurm que não estejam listados como negados em `CustomSlurmSettings`. Para obter mais informações sobre parâmetros de


configuração negados do Slurm, consulte [Parâmetros de configuração do Slurm listados como negados para CustomSlurmSettings](#).

AWS ParallelCluster só verifica se um parâmetro está em uma lista de negação. AWS ParallelCluster não valida a sintaxe ou a Slurm semântica do parâmetro de configuração personalizado. Você é responsável por validar seus parâmetros de configuração personalizados do Slurm. Parâmetros de configuração personalizados do Slurm inválidos podem causar falhas no daemon do Slurm que podem levar a falhas na criação e atualização do cluster.

Para obter mais informações sobre como especificar parâmetros Slurm de configuração personalizados com AWS ParallelCluster, consulte [Personalização de configuração do Slurm](#).

Para obter mais informações sobre os parâmetros de configuração Slurm, consulte [slurm.conf](#) na documentação do Slurm.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

 Note

CustomSlurmSettings é suportado a partir do AWS ParallelCluster versão 3.6.0.

Tags (Opcional, [String])

Uma lista de pares de tag com chave-valor. As tags [ComputeResource](#) substituem as tags duplicadas especificadas em [Seção Tags](#) ou em `SlurmQueues / Tags`.

Key (Opcional, **String**)

A chave de tags.

Value (Opcional, **String**)

O valor da tag.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

HealthChecks (Opcional)

Especifica as verificações de integridade do nó de computação em todos os recursos computacionais na fila.

Gpu (Opcional)

Especifica as verificações de integridade dos recursos de computação em uma fila.

Note

AWS ParallelCluster não suporta `HealthChecks/Gpu` em nós que usam sistemas operacionais `linux2 ARM`. Essas plataformas não são compatíveis com o [NVIDIA Data Center GPU Manager \(DCGM\)](#).

Enabled (Opcional, Boolean)

Se AWS ParallelCluster realiza verificações de integridade da GPU nos nós de computação. O padrão é `false`.

Comportamento de verificação de integridade de Gpu

- Se `Gpu / Enabled` estiver definido como `true`, AWS ParallelCluster executa verificações de integridade da GPU nos recursos de computação na fila.
- A verificação de integridade `Gpu` realiza verificações de integridade da GPU nos recursos de computação para evitar o envio de trabalhos em nós com uma GPU degradada.
- Se um nó de computação falhar em uma verificação de integridade do `Gpu`, o estado do nó de computação mudará para `DRAIN`. Novos trabalhos não começam nesse nó. Os trabalhos existentes são executados até a conclusão. Depois que todos os trabalhos em execução forem concluídos, o nó de computação será encerrado se for um nó dinâmico e será substituído se for um nó estático.
- A duração da verificação de integridade do `Gpu` depende do tipo de instância selecionada, do número de GPUs na instância e do número de destinos da verificação de integridade `Gpu` (equivalente ao número de destinos da GPU do trabalho). Para uma instância com 8 GPUs, a duração típica é inferior a 3 minutos.
- Se a verificação de integridade do `Gpu` for executada em uma instância sem suporte, ela será encerrada e o trabalho será executado no nó de computação. Por exemplo, se uma instância não tiver uma GPU, ou se uma instância tiver uma GPU, mas não for uma GPU NVIDIA, a verificação de integridade será encerrada e o trabalho será executado no nó de computação. Somente GPUs NVIDIA são suportadas.
- A verificação de integridade `Gpu` usa a ferramenta `dcgm` para realizar verificações de integridade em um nó e segue as seguintes etapas:


Quando a verificação de integridade Gpu começa em um nó:

1. Ele detecta se os serviços `nvidia-dcgm` e `nvidia-fabricmanager` estão em execução.
2. Se esses serviços não estiverem em execução, a verificação de integridade Gpu os iniciará.
3. Ele detecta se o modo de persistência está ativado.
4. Se o modo de persistência não estiver ativado, a verificação de integridade Gpu o ativará.

Ao final da verificação de saúde, a verificação de integridade Gpu restaura esses serviços e recursos ao estado inicial.

- Se o trabalho for atribuído a um conjunto específico de GPUs de nós, a verificação de integridade Gpu será executada somente nesse conjunto específico. Caso contrário, a verificação de integridade Gpu será executada em todas as GPUs do nó.
- Se um nó de computação receber duas ou mais solicitações de verificação de integridade Gpu ao mesmo tempo, somente a primeira verificação de integridade será executada e as outras serão ignoradas. Esse também é o caso das verificações de integridade direcionadas às GPUs de nós. Você pode verificar os arquivos de log para obter informações adicionais sobre essa situação.
- O log de verificação de integridade de um nó de computação específico está disponível no arquivo `/var/log/parallelcluster/slurm_health_check.log`. O arquivo está disponível na Amazon CloudWatch, no grupo de CloudWatch logs do cluster, onde você pode encontrar:
 - Detalhes sobre a ação executada pela verificação de integridade Gpu, incluindo a ativação e desativação de serviços e o modo de persistência.
 - O identificador da GPU, o ID serial e o UUID.
 - A saída da verificação de integridade.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

 Note

HealthChecks é suportado a partir da AWS ParallelCluster versão 3.6.0.

Networking

(Obrigatório) Define a configuração de rede para a fila do Slurm.

```
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
  PlacementGroup:
    Enabled: boolean
    Id: string
    Name: string
  Proxy:
    HttpProxyAddress: string
```

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Propriedades do **Networking**

SubnetIds (Obrigatório, [**String**])

As IDs das sub-redes existentes nas quais você provisiona a fila Slurm.

Se você configurar os tipos de instância em [SlurmQueues](#) / [ComputeResources](#) / [InstanceType](#), só poderá definir uma sub-rede.

Se você configurar os tipos de instância em [SlurmQueues](#) / [ComputeResources](#) / [Instances](#), poderá definir uma única sub-rede ou várias sub-redes.

Se você usa várias sub-redes, todas as sub-redes definidas para uma fila devem estar na mesma VPC, com cada sub-rede em uma zona de disponibilidade (AZ) separada.


Por exemplo, suponha que você defina subnet-1 e subnet-2 para sua fila.

subnet-1 e subnet-2 não podem ambas estar no AZ-1.

subnet-1 pode estar em AZ-1 e subnet-2 pode estar em AZ-2.

Se você configurar somente um tipo de instância e quiser usar várias sub-redes, defina seu tipo de instância em `Instances` vez de `InstanceType`.

Por exemplo, defina `ComputeResources / Instances / InstanceType=instance.type` em vez de `ComputeResources / InstanceType=instance.type`.

 Note

O Elastic Fabric Adapter (EFA) não é compatível com diferentes zonas de disponibilidade.

O uso de várias zonas de disponibilidade pode causar aumentos na latência da rede de armazenamento e custos adicionais de transferência de dados entre AZ. Por exemplo, isso pode ocorrer quando uma instância acessa o armazenamento de arquivos localizado em uma AZ diferente. Para obter mais informações, consulte [Transferência de dados dentro da mesma Região da AWS](#).

Atualizações de cluster para mudar do uso de uma única sub-rede para várias sub-redes:


- Suponha que a definição de sub-rede de um cluster seja definida com uma única sub-rede e um sistema de arquivos FSx for Lustre AWS ParallelCluster gerenciado. Então, você não pode atualizar esse cluster diretamente com uma definição de ID de sub-rede atualizada. Para fazer a atualização do cluster, você deve primeiro alterar o sistema de arquivos gerenciado para um sistema de arquivos externo. Para ter mais informações, consulte [Converter armazenamento gerenciado do AWS ParallelCluster em armazenamento externo](#).
- Suponha que a definição de sub-rede de um cluster seja definida com uma única sub-rede e um sistema de arquivos externo do Amazon EFS se os destinos de montagem do EFS não existirem para todas as AZs das várias sub-redes definidas para serem adicionadas. Então, você não pode atualizar esse cluster diretamente com uma definição de ID de sub-rede atualizada. Para atualizar o cluster ou criar um cluster, você deve primeiro criar todos os destinos de montagem para todas as AZs das várias sub-redes definidas.

Zonas de disponibilidade e reservas de capacidade de cluster definidas em [CapacityReservationResourceGroupArn](#):


- Você não pode criar um cluster se não houver sobreposição entre o conjunto de tipos de instância e zonas de disponibilidade cobertas pelo grupo de recursos de reserva de capacidade definido e o conjunto de tipos de instância e zonas de disponibilidade definidos para a fila.
- Você pode criar um cluster se houver uma sobreposição parcial entre o conjunto de tipos de instância e zonas de disponibilidade cobertos pelo grupo de recursos de reserva de capacidade

definido e o conjunto de tipos de instância e zonas de disponibilidade definidos para a fila. AWS ParallelCluster envia uma mensagem de aviso sobre a sobreposição parcial desse caso.

- Para ter mais informações, consulte [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#).

 Note

Várias zonas de disponibilidade foram adicionadas na AWS ParallelCluster versão 3.4.0.

 Warning

Esse aviso se aplica a todas as AWS ParallelCluster versões 3.x.y anteriores à versão 3.3.1. AWS ParallelCluster a versão 3.3.1 não será afetada se esse parâmetro for alterado.

Para AWS ParallelCluster 3 versões anteriores à versão 3.3.1:

Alterar esse parâmetro e atualizar um cluster cria um novo sistema de arquivos FSx para Lustre gerenciado e exclui o sistema de arquivos FSx para Lustre gerenciado existente sem preservar os dados existentes. Isso resulta em perda de dados. Antes de continuar, certifique-se de fazer backup dos dados do sistema de arquivos FSx para Lustre existente se quiser preservar os dados. Para obter mais informações, consulte [Trabalhar com backups](#) no Guia do usuário do FSx para Lustre.

Se um novo valor de sub-rede for adicionado, [Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Se um valor de sub-rede for removido, [Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

AssignPublicIp (Opcional, **String**)

Cria ou atribui um endereço IP público aos nós na fila do Slurm. Os valores compatíveis são `true` e `false`. A sub-rede que você especifica determina o valor padrão. Uma sub-rede com IPs públicos usa como padrão a atribuição de endereços IP públicos.

Se você definir um tipo de hpc6id instância p4d ou, ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede, [ElasticIp](#) define

[HeadNodeNetworking](#)//como true para fornecer acesso público. AWS IPs públicos só podem ser atribuídos a instâncias iniciadas com uma única interface de rede. Nesse caso, recomendamos que você use um [gateway NAT](#) para fornecer acesso público aos nós de computação do cluster. Neste caso, defina `AssignPublicIp` para false. Para obter mais informações sobre endereços IP, consulte [Atribuir um endereço IPv4 público durante a execução de uma instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

SecurityGroups (Opcional, [String])

Uma lista de grupos de segurança a usar para a fila Slurm. Se nenhum grupo de segurança for especificado, AWS ParallelCluster cria grupos de segurança para você.

Verifique se os grupos de segurança estão configurados corretamente para seus [SharedStorage](#) sistemas.

Warning

Este aviso se aplica a todos os 3. x. y AWS ParallelCluster versões anteriores à versão 3.3.0. AWS ParallelCluster a versão 3.3.0 não será afetada se esse parâmetro for alterado.

Para AWS ParallelCluster 3 versões anteriores à versão 3.3.0:

Alterar esse parâmetro e atualizar um cluster cria um novo sistema de arquivos FSx para Lustre gerenciado e exclui o sistema de arquivos FSx para Lustre gerenciado existente sem preservar os dados existentes. Isso resulta em perda de dados. Certifique-se de fazer backup dos dados do sistema de arquivos FSx para Lustre existente se quiser preservar os dados. Para obter mais informações, consulte [Trabalhar com backups](#) no Guia do usuário do FSx para Lustre.

Warning

Se você habilitar o [Efa](#) para suas instâncias de computação, certifique-se de que suas instâncias habilitadas para EFA sejam membros de um grupo de segurança que permita todo o tráfego de entrada e saída para si.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

AdditionalSecurityGroups (Opcional, **[String]**)

Uma lista adicional de grupos de segurança a usar para a fila Slurm.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

PlacementGroup (Opcional)

Especifica as configurações do grupo de posicionamento para a fila Slurm.

```
PlacementGroup:  
  Enabled: boolean  
  Id: string  
  Name: string
```

[Política de atualização: todos os nós de computação devem ser interrompidos para a exclusão de um grupo de posicionamento gerenciado. A frota de computação deve ser interrompida ou QueueUpdateStrategy deve ser configurada para que essa configuração seja alterada para uma atualização.](#)

Enabled (Opcional, **Boolean**)

Indica se um grupo de posicionamento é usado para a fila Slurm. O padrão é `false`.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Id (Opcional, **String**)

O nome do grupo de posicionamento de um grupo de posicionamento de cluster existente que a fila Slurm usa. Certifique-se de fornecer o nome do grupo de posicionamento e não o ID.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Name (Opcional, **String**)

O nome do grupo de posicionamento de um grupo de posicionamento de cluster existente que a fila Slurm usa. Certifique-se de fornecer o nome do grupo de posicionamento e não o ID.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Note

- Se `PlacementGroup / Enabled` for definido como `true`, sem um `Name` ou `Id` definido, cada recurso de computação receberá seu próprio grupo de posicionamento gerenciado, a menos que [ComputeResources / Networking / PlacementGroup](#) seja definido para substituir essa configuração.
- A partir da AWS ParallelCluster versão 3.3.0, [SlurmQueues/Networking/PlacementGroup/Name](#) foi adicionado como alternativa preferencial para [SlurmQueues//NetworkingPlacementGroup/Id](#).

[PlacementGroup / Id](#) e [PlacementGroup / Name](#) são equivalentes. Você pode usar qualquer um.

Se você incluir [PlacementGroup/Id](#) e [PlacementGroup/Name](#), AWS ParallelCluster falhará. É preciso escolher um ou outro.

Você não precisa atualizar seu cluster para usar [PlacementGroup / Name](#).

Proxy (Opcional)

Especifica as configurações de proxy para o a fila Slurm.

```
Proxy:  
  HttpProxyAddress: string
```

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

HttpProxyAddress (Opcional, **String**)

Define um servidor proxy HTTP ou HTTPS para a fila Slurm. Normalmente, é `https://x.x.x.x:8080`.

Não há valor padrão.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Image

(Opcional) Especifica a imagem a ser usada na fila Slurm. Para usar a mesma AMI em todos os nós, use a [CustomAmi](#) configuração na [Imagem](#) seção.

```
Image:  
  CustomAmi: string
```

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Propriedades do **Image**

CustomAmi (Opcional, **String**)

A AMI a ser usada para a fila Slurm em vez das AMIs padrão. Você pode usar o comando CLI do `pcluster` para ver uma lista das AMIs padrão.

Note

A AMI deve ser baseada no mesmo sistema operacional usado pelo nó principal.

```
pcluster list-official-images
```

Se a AMI personalizada exigir permissões adicionais para sua execução, você deverá adicionar essas permissões à política do nó principal.

Por exemplo, se uma AMI personalizada tiver um snapshot criptografado associado a ela, as seguintes políticas adicionais serão necessárias nas políticas do usuário e do nó principal.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:DescribeKey",  
        "kms:ReEncrypt*"  
      ]  
    }  
  ]  
}
```

```

        "kms:CreateGrant",
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
    ]
}
]
}

```

Para solucionar problemas de avisos de validação da AMI personalizada, consulte [Solução de problemas de AMI personalizada](#).

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy](#) para que essa configuração seja alterada para uma atualização.

ComputeResources

(Obrigatório) Define a configuração ComputeResources para a fila Slurm.

Note

O tamanho do cluster pode mudar durante uma atualização. Para obter mais informações, consulte [Capacidade, tamanho e atualização do 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
```

Política de atualização: para essa configuração de valores de lista, um novo valor pode ser adicionado durante uma atualização ou a frota de computação deve ser interrompida ao remover um valor existente.

Propriedades do **ComputeResources**

Name (Obrigatório, **String**)

O nome do ambiente de computação da fila do Slurm. O nome do pode ter até 25 caracteres.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

InstanceType (Obrigatório, **String**)

O tipo de instância usado nesse recurso de computação Slurm. Todos os tipos de instância em um cluster devem usar a mesma arquitetura de processador. As instâncias podem usar a arquitetura x86_64 ou arm64.

A configuração do cluster deve definir uma [InstanceType](#) ou duas [Instâncias](#). Se ambos estiverem definidos, AWS ParallelCluster falhará.

Ao definir InstanceType, você não pode definir várias sub-redes. Se você configurar somente um tipo de instância e quiser usar várias sub-redes, defina seu tipo de instância em [Instances](#) em vez de InstanceType. Para obter mais informações, consulte [Networking](#) / [SubnetIds](#).

Se você definir um tipo de hpc6id instância p4d ou, ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede, deverá executar as instâncias

de computação na sub-rede privada, conforme descrito em. [O AWS ParallelCluster com duas sub-redes](#) AWS IPs públicos só podem ser atribuídos a instâncias que são iniciadas com uma única interface de rede. Para obter mais informações, consulte [Atribuir um endereço IPv6 a uma instância](#), no Guia do usuário do Amazon EC2 para instâncias Linux.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

Instances (Obrigatório)

Especifica a lista de tipos de instância para um recurso de computação. Para especificar a estratégia de alocação para a lista de tipos de instância, consulte [AllocationStrategy](#).

A configuração do cluster deve definir [InstanceType](#) ou [Instances](#). Se ambos estiverem definidos, AWS ParallelCluster falhará.

Para ter mais informações, consulte [Alocação a vários tipos de instância com o Slurm](#).

[Instances](#):

- [InstanceType](#): *string*

Note

[A partir da AWS ParallelCluster versão 3.7.0, EnableMemoryBasedScheduling pode ser ativado se você configurar vários tipos de instância em Instâncias.](#)

Para AWS ParallelCluster as versões 3.2.0 a 3.6. x, não

EnableMemoryBasedScheduling pode ser ativado se você configurar vários tipos de instância em [Instâncias](#).

Política de atualização: para essa configuração de valores de lista, um novo valor pode ser adicionado durante uma atualização ou a frota de computação deve ser interrompida ao remover um valor existente.

InstanceType (Obrigatório, **String**)

O tipo de instância usado nesse recurso de computação Slurm. Todos os tipos de instância em um cluster devem usar a mesma arquitetura de processador, seja x86_64 ou arm64.


Os tipos de instância listados em [Instances](#) devem ter:

- O mesmo número de vCPUs ou se [DisableSimultaneousMultithreading](#) for definido como true, o mesmo número de núcleos.
- O mesmo número de aceleradores dos mesmos fabricantes.
- Compatível com EFA, se [Efa](#) / [Enabled](#) definido como true.

Os tipos de instância listados em [Instances](#) devem ter:

- Quantidade diferente de memória.

Nesse caso, a memória mínima deve ser definida como um recurso consumível Slurm.

 Note

[A partir da AWS ParallelCluster versão 3.7.0, EnableMemoryBasedScheduling pode ser ativado se você configurar vários tipos de instância em Instâncias.](#)

Para AWS ParallelCluster as versões 3.2.0 a 3.6. x, não [EnableMemoryBasedScheduling](#) pode ser ativado se você configurar vários tipos de instância em [Instâncias](#).

- Placas de rede diferentes.

Nesse caso, o número de interfaces de rede configuradas para o recurso computacional é definido pelo tipo de instância com o menor número de placas de rede.

- Largura de banda da rede diferente.
- Tamanho diferente do armazenamento de instâncias.

Se você definir um tipo de hpc6id instância p4d ou, ou outro tipo de instância que tenha várias interfaces de rede ou uma placa de interface de rede, deverá executar as instâncias de computação na sub-rede privada, conforme descrito em [O AWS ParallelCluster com duas sub-redes](#) AWS IPs públicos só podem ser atribuídos a instâncias iniciadas com uma única interface de rede. Para obter mais informações, consulte [Atribuir um endereço IPv6 a uma instância](#), no Guia do usuário do Amazon EC2 para instâncias Linux.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

 Note

Instances é suportado a partir da AWS ParallelCluster versão 3.3.0.

MinCount (Opcional, Integer)

O número mínimo de instâncias que o recurso de computação Slurm usa. O padrão é 0.

Note

O tamanho do cluster pode mudar durante uma atualização. Para obter mais informações, consulte [Capacidade, tamanho e atualização do cluster](#)

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

MaxCount (Opcional, Integer)

O número máximo de instâncias que o recurso de computação Slurm usa. O padrão é 10.

Ao usar `CapacityType = CAPACITY_BLOCK`, `MaxCount` deve ser igual `MinCount` e maior que 0, porque todas as instâncias que fazem parte da reserva do Bloco de Capacidade são gerenciadas como nós estáticos.

No momento da criação do cluster, o nó principal espera que todos os nós estáticos estejam prontos antes de sinalizar o sucesso da criação do cluster. No entanto, ao usar `CapacityType = CAPACITY_BLOCK`, os nós que fazem parte dos recursos computacionais associados aos blocos de capacidade não serão considerados para essa verificação. O cluster será criado mesmo que nem todos os blocos de capacidade configurados estejam ativos.

Note

O tamanho do cluster pode mudar durante uma atualização. Para obter mais informações, consulte [Capacidade, tamanho e atualização do cluster](#)

DynamicNodePriority (Opcional, Integer)

A prioridade dos nós dinâmicos em um recurso de computação em fila. A prioridade é mapeada para o parâmetro de configuração `Weight` do nó do Slurm para os nós dinâmicos do recurso de computação. O valor padrão é 1000.

Slurm prioriza primeiro os nós com os valores de `Weight` mais baixos.

⚠ Warning

O uso de muitos valores `Weight` diferentes em uma partição Slurm (fila) pode diminuir a taxa de agendamento de trabalhos na fila.

Nas AWS ParallelCluster versões anteriores à versão 3.7.0, os nós estáticos e dinâmicos receberam o mesmo peso padrão de 1. Nesse caso, Slurm pode priorizar nós dinâmicos ociosos sobre nós estáticos ociosos devido ao esquema de nomenclatura para nós estáticos e dinâmicos. Quando tudo estiver igual, o Slurm programa os nós em ordem alfabética por nome.

ℹ Note

`DynamicNodePriority` é adicionado na AWS ParallelCluster versão 3.7.0.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

StaticNodePriority (Opcional, **Integer**)

A prioridade dos nós dinâmicos em um recurso de computação em fila. A prioridade é mapeada para o parâmetro de configuração `Weight` do nó do Slurm para os nós estáticos do recurso de computação. O valor padrão é 1.

Slurm prioriza primeiro os nós com os valores de `Weight` mais baixos.

⚠ Warning

O uso de muitos valores `Weight` diferentes em uma partição Slurm (fila) pode diminuir a taxa de agendamento de trabalhos na fila.

ℹ Note

`StaticNodePriority` é adicionado na AWS ParallelCluster versão 3.7.0.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

SpotPrice (Opcional, **Float**)

O preço máximo pago por uma instância spot do EC2 antes de qualquer instância ser lançada. O valor padrão é o preço sob demanda.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

DisableSimultaneousMultithreading (Opcional, **Boolean**)

Se `true`, o multithreading nos nós da fila Slurm estiver desativado. O valor padrão é `false`.

Nem todos os tipos de instância podem desativar o multithreading. Para obter uma lista de tipos de instância que suportam a desativação do multithreading, consulte [Núcleos e threads de CPU para cada núcleo de CPU para cada tipo de instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

SchedulableMemory (Opcional, **Integer**)

A quantidade de memória em MiB configurada no parâmetro `RealMemory` do Slurm para os nós de computação de um recurso de computação. Esse valor é o limite superior da memória do nó disponível para trabalhos quando [SlurmSettings / EnableMemoryBasedScheduling](#) está habilitado. O valor padrão é 95 por cento da memória listada nos [tipos de instância do Amazon EC2](#) e retornada pela API do Amazon EC2. [DescribeInstanceTypes](#) Certifique-se de converter valores fornecidos em GiB em MiB.

Valores com suporte: 1-`EC2Memory`

`EC2Memory` é a memória (em MiB) listada nos tipos de [instância do Amazon EC2](#) e retornada pela API do Amazon EC2. [DescribeInstanceTypes](#) Certifique-se de converter valores fornecidos em GiB em MiB.

Essa opção é mais relevante quando [SlurmSettings / EnableMemoryBasedScheduling](#) está habilitada. Para ter mais informações, consulte [Programação baseada em memória do Slurm](#).

Note

`SchedulableMemory` é suportado a partir do AWS ParallelCluster versão 3.2.0.

A partir da versão 3.2.0, por padrão, AWS ParallelCluster configura os `RealMemory` nós de Slurm computação em 95% da memória que é retornada pela API do Amazon EC2. `DescribeInstanceTypes` Essa configuração é independente do valor de `EnableMemoryBasedScheduling`.

[Política de atualização: a frota de computação deve ser interrompida ou `QueueUpdateStrategy` para que essa configuração seja alterada para uma atualização.](#)

HealthChecks (Opcional)

Especifica as verificações de integridade em um recurso de computação.

Gpu (Opcional)

Especifica as verificações de integridade em um recurso de computação.

Enabled (Opcional, Boolean)

Se AWS ParallelCluster realiza verificações de integridade da GPU na computação de um recurso em uma fila. O padrão é `false`.

Note

AWS ParallelCluster não suporta `HealthChecks/Gpu` em nós que usam sistemas operacionais `alinux2 ARM`. Essas plataformas não são compatíveis com o [NVIDIA Data Center GPU Manager \(DCGM\)](#).

Comportamento de verificação de integridade de Gpu

- Se `Gpu/Enabled` estiver definido como `true`, AWS ParallelCluster executa verificações de integridade da GPU em um recurso computacional.
- A verificação de integridade do Gpu realiza verificações de integridade em um recurso de computação para evitar o envio de trabalhos em nós com uma GPU degradada.
- Se um nó de computação falhar em uma verificação de integridade do Gpu, o estado do nó de computação mudará para `DRAIN`. Novos trabalhos não começam nesse nó. Os trabalhos existentes são executados até a conclusão. Depois que todos os trabalhos em execução forem concluídos, o nó de computação será encerrado se for um nó dinâmico e será substituído se for um nó estático.

- A duração da verificação de integridade do Gpu depende do tipo de instância selecionada, do número de GPUs na instância e do número de destinos da verificação de integridade Gpu (equivalente ao número de destinos da GPU do trabalho). Para uma instância com 8 GPUs, a duração típica é inferior a 3 minutos.
- Se a verificação de integridade do Gpu for executada em uma instância sem suporte, ela será encerrada e o trabalho será executado no nó de computação. Por exemplo, se uma instância não tiver uma GPU, ou se uma instância tiver uma GPU, mas não for uma GPU NVIDIA, a verificação de integridade será encerrada e o trabalho será executado no nó de computação. Somente GPUs NVIDIA são suportadas.
- A verificação de integridade Gpu usa a ferramenta `dcgmi` para realizar verificações de integridade em um nó e segue as seguintes etapas:

Quando a verificação de integridade Gpu começa em um nó:

1. Ele detecta se os serviços `nvidia-dcgm` e `nvidia-fabricmanager` estão em execução.
2. Se esses serviços não estiverem em execução, a verificação de integridade Gpu os iniciará.
3. Ele detecta se o modo de persistência está ativado.
4. Se o modo de persistência não estiver ativado, a verificação de integridade Gpu o ativará.

Ao final da verificação de saúde, a verificação de integridade Gpu restaura esses serviços e recursos ao estado inicial.

- Se o trabalho for atribuído a um conjunto específico de GPUs de nós, a verificação de integridade Gpu será executada somente nesse conjunto específico. Caso contrário, a verificação de integridade Gpu será executada em todas as GPUs do nó.
- Se um nó de computação receber duas ou mais solicitações de verificação de integridade Gpu ao mesmo tempo, somente a primeira verificação de integridade será executada e as outras serão ignoradas. Esse também é o caso das verificações de integridade direcionadas às GPUs de nós. Você pode verificar os arquivos de log para obter informações adicionais sobre essa situação.
- O log de verificação de integridade de um nó de computação específico está disponível no arquivo `/var/log/parallelcluster/slurm_health_check.log`. Esse arquivo está disponível na Amazon CloudWatch, no grupo de CloudWatch logs do cluster, onde você pode encontrar:

- Detalhes sobre a ação executada pela verificação de integridade Gpu, incluindo a ativação e desativação de serviços e o modo de persistência.
- O identificador da GPU, o ID serial e o UUID.
- A saída da verificação de integridade.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

HealthChecks é suportado a partir da AWS ParallelCluster versão 3.6.0.

Efa (Opcional)

Especifica as configurações do Elastic Fabric Adapter (EFA) para os nós na fila Slurm.

Efa:

Enabled: *boolean*

GdrSupport: *boolean*

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Enabled (Opcional, Boolean)

Especifica que o Elastic Fabric Adapter (EFA) está ativado. Para visualizar uma lista de tipos de instâncias do EC2 compatíveis com EFAs, consulte [Tipos de instância compatíveis](#) no Guia do usuário do Amazon EC2 para instâncias do Linux. Para ter mais informações, consulte [Elastic Fabric Adapter](#). Recomendamos que você use um cluster [SlurmQueues / Networking / PlacementGroup](#) para minimizar as latências entre as instâncias.

O valor padrão é `false`.

Note

O Elastic Fabric Adapter (EFA) não é compatível com diferentes zonas de disponibilidade. Para obter mais informações, consulte [SubnetIds](#).

⚠ Warning

Se você estiver definindo um grupo de segurança personalizado em [SecurityGroups](#), certifique-se de que suas instâncias habilitadas para EFA sejam membros de um grupo de segurança que permite todo o tráfego de entrada e saída para si mesmo.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

GdrSupport (Opcional, **Boolean**)

(Opcional) A partir do AWS ParallelCluster versão 3.0.2, essa configuração não tem efeito. O suporte do Elastic Fabric Adapter (EFA) para GPUDirect RDMA (acesso direto remoto à memória) está sempre ativado se for suportado pelo tipo de instância do recurso de computação Slurm e do sistema operacional.

ℹ Note

AWS ParallelCluster versão 3.0.0 a 3.0.1: O suporte para GPUDirect RDMA está habilitado para recursos de computação. Slurm O suporte para GPUDirect RDMA é compatível com tipos de instância específicos (p4d.24xlarge) em sistemas operacionais específicos ([Os](#) é `alinux2`, `centos7`, `ubuntu1804` ou `ubuntu2004`). O valor padrão é falso.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

CapacityReservationTarget

[CapacityReservationTarget](#):

[CapacityReservationId](#): *string*

[CapacityReservationResourceGroupArn](#): *string*

Especifica a reserva de capacidade sob demanda para usar com os recursos de computação da fila.

CapacityReservationId (Opcional, **String**)

O ID da reserva de capacidade existente a ser direcionada para os recursos de computação da fila. O id pode se referir a um [ODCR](#) ou a um [bloco de capacidade para ML](#).

Quando esse parâmetro for especificado no nível do recurso computacional, InstanceType for opcional, ele será automaticamente recuperado da reserva.

CapacityReservationResourceGroupArn (Opcional, **String**)

Indica o nome do recurso da Amazon (ARN) do grupo de recursos que serve como o grupo de reservas de capacidade vinculadas ao serviço do recurso de computação. AWS ParallelCluster identifica e usa a reserva de capacidade mais adequada ao grupo. O grupo de recursos deve ter pelo menos um ODCR para cada tipo de instância listado para o recurso de computação. Para ter mais informações, consulte [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#).

- Se PlacementGroup estiver habilitado em [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleciona um grupo de recursos que tem como alvo o tipo de instância e PlacementGroup um recurso computacional, se ele existir.

PlacementGroup precisa ter como alvo um dos tipos de instância definidos em [ComputeResources](#).

- Se PlacementGroup não estiver habilitado em [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster seleciona um grupo de recursos que tem como alvo somente o tipo de instância de um recurso computacional, se ele existir.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Note

CapacityReservationTarget é adicionado com o AWS ParallelCluster versão 3.3.0.

Networking

[Networking](#):
[PlacementGroup](#):

Enabled: *boolean*

Name: *string*

Política de atualização: todos os nós de computação devem ser interrompidos para a exclusão de um grupo de posicionamento gerenciado. A frota de computação deve ser interrompida ou QueueUpdateStrategy deve ser configurada para que essa configuração seja alterada para uma atualização.

PlacementGroup (Opcional)

Especifica as configurações do grupo de posicionamento para o recurso de computação.

Enabled (Opcional, Boolean)

Indica se um grupo de posicionamento é usado para o recurso de computação.

- Se definido como `true`, sem um `Name` definido, esse recurso de computação receberá seu próprio grupo de posicionamento gerenciado, independentemente da configuração [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#).
- Se definido como `true`, sem um `Name` definido, esse recurso de computação receberá seu próprio grupo de posicionamento nomeado, independentemente das configurações [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#).

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Name (Opcional, String)

O nome do grupo de posicionamento de um grupo de posicionamento de cluster existente usado pelo recurso de computação.

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Note

- Se `PlacementGroup` / `Enabled` e `Name` não estiverem definidos, seus respectivos valores serão padronizados para as configurações [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#).
- `ComputeResources/Networking/PlacementGroup` é adicionado com a AWS ParallelCluster versão 3.3.0.

CustomSlurmSettings (Opcional, **Dict**)

(Opcional) Define as configurações personalizadas do nó Slurm (recurso de computação).

Especifica um dicionário de pares de valores-chave de parâmetros de configuração Slurm personalizados que se aplicam aos nós Slurm (recursos de computação).

Cada par de valores-chave separado, como `Param1: Value1`, é adicionado separadamente ao final da linha de configuração do nó Slurm no formato `Param1=Value1`.

Você só pode especificar parâmetros de configuração Slurm que não estejam listados como negados em `CustomSlurmSettings`. Para obter mais informações sobre parâmetros de configuração negados do Slurm, consulte [Parâmetros de configuração do Slurm listados como negados para CustomSlurmSettings](#).

AWS ParallelCluster só verifica se um parâmetro está em uma lista de negação. AWS ParallelCluster não valida a sintaxe ou a Slurm semântica do parâmetro de configuração personalizado. Você é responsável por validar seus parâmetros de configuração personalizados do Slurm. Parâmetros de configuração personalizados do Slurm inválidos podem causar falhas no daemon do Slurm que podem levar a falhas na criação e atualização do cluster.

Para obter mais informações sobre como especificar parâmetros Slurm de configuração personalizados com AWS ParallelCluster, consulte [Personalização de configuração do Slurm](#).

Para obter mais informações sobre os parâmetros de configuração Slurm, consulte [slurm.conf](#) na documentação do Slurm.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

`CustomSlurmSettings` é suportado a partir do AWS ParallelCluster versão 3.6.0.

Tags (Opcional, [String])

Uma lista de pares de tag com chave-valor. As tags `ComputeResource` substituem as tags duplicadas especificadas em [Seção Tags](#) ou [SlurmQueues](#) / Tags.

Key (Opcional, **String**)

A chave de tags.

Value (Opcional, **String**)

O valor da tag.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

ComputeSettings

(Obrigatório) Define a configuração ComputeSettings para a fila Slurm.

Propriedades do **ComputeSettings**

Especifica as propriedades ComputeSettings dos nós na fila Slurm.

```
ComputeSettings:  
LocalStorage:  
  RootVolume:  
    Size: integer  
    Encrypted: boolean  
    VolumeType: string  
    Iops: integer  
    Throughput: integer  
  EphemeralVolume:  
    MountDir: string
```

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

LocalStorage (Opcional)

Especifica as propriedades LocalStorage dos nós na fila Slurm.

```
LocalStorage:  
  RootVolume:  
    Size: integer  
    Encrypted: boolean  
    VolumeType: string  
    Iops: integer  
    Throughput: integer  
  EphemeralVolume:  
    MountDir: string
```


[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

RootVolume (Opcional)

Especifica os detalhes do volume raiz dos nós na fila Slurm.

RootVolume:

Size: *integer*

Encrypted: *boolean*

VolumeType: *string*

Iops: *integer*

Throughput: *integer*

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Size (Opcional, **Integer**)

Especifica o tamanho do volume raiz em gibibytes (GiB) para os nós na fila Slurm. O tamanho padrão vem da AMI. Usar um tamanho diferente exige que AMI seja compatível com growroot.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Encrypted (Opcional, **Boolean**)

Se true, o volume raiz dos nós na fila Slurm ficam criptografados. O valor padrão é false.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

VolumeType (Opcional, **String**)

Especifica o [tipo de volume do Amazon EBS](#) dos nós na fila Slurm. Os valores compatíveis são gp2, gp3, io1, io2, sc1, st1 e standard. O valor padrão é gp3.

Para ter mais informações, consulte [Tipos de volumes do Amazon EBS](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Iops (Opcional, **Boolean**)

Define o número de IOPS para os volumes do tipo `io1`, `io2` e `gp3`.

O valor padrão, os valores suportados e a proporção do `volume_iops` para `volume_size` variam de acordo com `VolumeType` e `Size`.

VolumeType = io1

Padrão Iops = 100

Valores suportados Iops = 100–64000 †

Proporção máxima de `volume_iops` para `volume_size` = 50 IOPS por GiB. 5000 IOPS exigem um `volume_size` de pelo menos 100 GiB.

VolumeType = io2

Padrão Iops = 100

Valores suportados Iops = 100–64000 (256000 para volumes do `io2 Block Express`) †

Proporção máxima de Iops para `Size` = 500 IOPS por GiB. 5000 IOPS exigem um `Size` de pelo menos 10 GiB.

VolumeType = gp3

Padrão Iops = 3000

Valores Iops suportados = 3000–16000 †

Relação de Iops máxima `Size` = 500 IOPS por GiB para volumes com IOPS maior que 3000.

† O número máximo de IOPS é garantido somente em [instâncias criadas no Nitro System](#) que também são provisionadas com mais de 32.000 IOPS. Outras instâncias podem ter até 32.000 IOPS. Os volumes de `io1` mais antigos podem não atingir o desempenho total, a menos que você [modifique o volume](#). `io2` Os volumes do `Block Express` oferecem suporte a valores `volume_iops` de até 256000 em tipos de instância `R5b`. Para obter mais informações, consulte [Volumes do io2 Block Express](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

[Política de atualização: a frota de computação deve ser interrompida ou `QueueUpdateStrategy` para que essa configuração seja alterada para uma atualização.](#)

Throughput (Opcional, **Integer**)

Define o throughput para tipos de volume gp3, em MiB/s. Isso é válido somente quando `VolumeType` é gp3. O valor padrão é 125. Valores suportados: 125–1000 MiB/s

A proporção de Throughput para Iops não pode ser superior a 0,25. O throughput máximo de 1000 MiB/s exige que a configuração Iops seja de pelo menos 4000.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

EphemeralVolume (Opcional, **Boolean**)

Especifica as configurações do volume efêmero. O volume efêmero é criado combinando todos os volumes de armazenamento de instâncias em um único volume lógico formatado com o sistema de arquivos ext4. O padrão é `/scratch`. Se o tipo de instância não tiver volumes de armazenamento de instâncias, nenhum volume efêmero será criado. Para obter mais informações, consulte [Volumes de armazenamento de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

```
EphemeralVolume:  
MountDir: string
```

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

MountDir (Opcional, **String**)

O diretório de montagem do volume efêmero de cada nó na fila do Slurm.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

CustomActions

(Opcional) Especifica scripts personalizados a serem executados nos nós da fila Slurm.

```
CustomActions:  
  OnNodeStart:  
    Sequence:  
      - Script: string  
      Args:
```

```
- string  
Script: string  
Args:  
  - string  
OnNodeConfigured:  
  Sequence:  
    - Script: string  
      Args:  
        - string  
  Script: string  
  Args:  
    - string
```

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Propriedades do **CustomActions**

OnNodeStart (Opcional, **String**)

Especifica uma sequência de scripts ou um único script a ser executado nos nós da fila Slurm antes que qualquer ação de bootstrap de implantação do nó seja iniciada. AWS ParallelCluster não suporta a inclusão de um único script e Sequence da mesma ação personalizada. Para ter mais informações, consulte [Ações de bootstrap personalizadas](#).

Sequence (Opcional)

Lista de scripts a serem executados.

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Script (Obrigatório, **String**)

O arquivo a ser usado. O caminho do arquivo pode começar com `https://` ou `s3://`.

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Args (Opcional, **[String]**)

Lista de argumentos a serem transmitidos para o script.

Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.

Script (Obrigatório, **String**)

O arquivo a ser usado para um único script. O caminho do arquivo pode começar com `https://` ou `s3://`.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Args (Opcional, **[String]**)

Lista de argumentos a serem transmitidos para o único script.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

OnNodeConfigured (Opcional, **String**)

Especifica uma sequência de scripts ou um único script a ser executado nos nós da fila Slurm após a conclusão de todas as ações de bootstrap do nó. AWS ParallelCluster não suporta a inclusão de um único script e Sequence da mesma ação personalizada. Para ter mais informações, consulte [Ações de bootstrap personalizadas](#).

Sequence (Opcional)

Lista de scripts a serem executados.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Script (Obrigatório, **String**)

O arquivo a ser usado. O caminho do arquivo pode começar com `https://` ou `s3://`.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Args (Opcional, **[String]**)

Lista de argumentos a serem transmitidos para o script.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Script (Obrigatório, **String**)

O arquivo a ser usado para um único script. O caminho do arquivo pode começar com `https://` ou `s3://`.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Args (Opcional, **[String]**)

Lista de argumentos a serem transmitidos para o script.

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

Note

Sequence é adicionado a partir da AWS ParallelCluster versão 3.6.0. Ao especificar `Sequence`, você pode listar vários scripts para uma ação personalizada. AWS ParallelCluster continua oferecendo suporte à configuração de uma ação personalizada com um único script, sem `Sequence`.

AWS ParallelCluster não suporta a inclusão de um único script e `Sequence` da mesma ação personalizada.

Iam

(Opcional) Define as configurações opcionais do IAM para a fila Slurm.

```
Iam:  
  S3Access:  
    - BucketName: string  
      EnableWriteAccess: boolean  
      KeyName: string  
  AdditionalIamPolicies:  
    - Policy: string  
  InstanceProfile: string  
  InstanceRole: string
```

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Propriedades do **Iam**

InstanceProfile (Opcional, **String**)

Especifica um perfil de instância para substituir a função de instância padrão ou o perfil de instância da fila Slurm. Não é possível especificar ambos InstanceProfile e InstanceRole. O formato é `arn:${Partition}:iam::${Account}:instance-profile/${InstanceProfileName}`.

Se isso for especificado, as configurações S3Access e AdditionalIamPolicies não poderão ser especificadas.

Recomendamos que você especifique uma ou ambas configurações S3Access e AdditionalIamPolicies, pois os recursos adicionados ao AWS ParallelCluster geralmente exigem novas permissões.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

InstanceRole (Opcional, **String**)

Especifica um perfil de instância para substituir a função de instância padrão ou o perfil de instância da fila Slurm. Não é possível especificar ambos InstanceProfile e InstanceRole. O formato é `arn:${Partition}:iam::${Account}:role/${RoleName}`.

Se isso for especificado, as configurações S3Access e AdditionalIamPolicies não poderão ser especificadas.

Recomendamos que você especifique uma ou ambas configurações S3Access e AdditionalIamPolicies, pois os recursos adicionados ao AWS ParallelCluster geralmente exigem novas permissões.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

S3Access (Opcional)

Especifica um bucket para a fila Slurm. Isso é usado para gerar políticas para conceder o acesso especificado ao bucket na fila Slurm.

Se isso for especificado, as configurações InstanceProfile e InstanceRole não poderão ser especificadas.

Recomendamos que você especifique uma ou ambas configurações `S3Access` e `AdditionalIamPolicies`, pois os recursos adicionados ao AWS ParallelCluster geralmente exigem novas permissões.

`S3Access`:

- `BucketName`: *string*
- `EnableWriteAccess`: *boolean*
- `KeyName`: *string*

Política de atualização: essa configuração pode ser alterada durante uma atualização.

BucketName (Obrigatório, **String**)

O nome do bucket do .

Política de atualização: essa configuração pode ser alterada durante uma atualização.

KeyName (Opcional, **String**)

A chave para o bucket. O valor padrão é *.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

EnableWriteAccess (Opcional, **Boolean**)

Indica se o acesso de gravação está habilitado para o bucket.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

AdditionalIamPolicies (Opcional)

Especifica uma lista de nomes de recursos da Amazon (ARNs) de políticas do IAM para o Amazon EC2. Essa lista é anexada à função raiz usada para a Slurm fila, além das permissões exigidas pelo AWS ParallelCluster.

O nome da política do IAM e seu ARN são diferentes. Os nomes não podem ser usados.

Se isso for especificado, as configurações `InstanceProfile` e `InstanceRole` não poderão ser especificadas.

Recomendamos usar `AdditionalIamPolicies` porque `AdditionalIamPolicies` são adicionadas às permissões exigidas pelo AWS ParallelCluster , e `InstanceRole` deve incluir todas as permissões necessárias. As permissões necessárias muitas vezes são alteradas entre as versões à medida que recursos são adicionados.

Não há valor padrão.

AdditionalIamPolicies:

- Policy: *string*

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Policy (Obrigatório, **[String]**)

Lista de políticas do IAM.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

SlurmSettings

(Opcional) Define as configurações para Slurm que se aplicam a todo o cluster.

SlurmSettings:

ScaledownIdleTime: *integer*

QueueUpdateStrategy: *string*

EnableMemoryBasedScheduling: *boolean*

CustomSlurmSettings: *[dict]*

CustomSlurmSettingsIncludeFile: *string*

Database:

Uri: *string*

UserName: *string*

PasswordSecretArn: *string*

Dns:

DisableManagedDns: *boolean*

HostedZoneId: *string*

UseEc2Hostnames: *boolean*

Propriedades do SlurmSettings

ScaledownIdleTime (Opcional, Integer)

Define por quanto tempo (em minutos) não há trabalho e o nó do Slurm que é encerrado.

O valor padrão é 10.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

MungeKeySecretArn (Opcional, **String**)

O Amazon Resource Name (ARN) do segredo em texto simples do Secrets AWS Manager que contém a chave munge codificada em base64 para ser usada no cluster Slurm. Essa chave munge será usada para autenticar chamadas RPC entre os comandos do cliente Slurm e os daemons do Slurm que atuam como servidores remotos. Se não MungeKeySecretArn for fornecido, AWS ParallelCluster gerará uma chave munge aleatória para o cluster.

Note

MungeKeySecretArn é suportado a partir da AWS ParallelCluster versão 3.8.0.

Warning

Se o MungeKeySecretArn for adicionado recentemente a um cluster existente, não ParallelCluster restaurará a chave munge anterior no caso de uma reversão ou ao remover posteriormente o MungeKeySecretArn. Em vez disso, uma nova chave munge aleatória será gerada.

Se o AWS ParallelCluster usuário tiver permissão para acessar [DescribeSecrets](#) esse recurso secreto específico, MungeKeySecretArn é validado. MungeKeySecretArn é válido se:

- O segredo especificado existe e
- O segredo é texto simples e contém uma string válida codificada em base64, e
- A chave munge binária decodificada tem um tamanho entre 256 e 8192 bits.

Se a política do IAM do usuário do pcluster não incluir DescribeSecret, ela não MungeKeySecretArn será validada e uma mensagem de aviso será exibida. Para ter mais informações, consulte [Política básica de usuário pcluster do AWS ParallelCluster](#).

Quando você atualiza MungeKeySecretArn, a frota computacional e todos os nós de login devem ser interrompidos.

Se o valor secreto no ARN secreto for modificado enquanto o ARN permanecer o mesmo, o cluster não será atualizado automaticamente com a nova chave munge. Para usar a nova chave munge do ARN secreto, você deve interromper a frota de computação e os nós de login e, em seguida, executar o seguinte comando a partir do nó principal.

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

Depois de executar o comando, você pode retomar a frota de computação e os nós de login: os nós de computação e login recém-provisionados começarão automaticamente a usar a nova chave munge.

Para gerar uma chave munge personalizada codificada em base64, você pode usar o utilitário [mungekey distribuído com o software munge e depois codificá-la usando o utilitário base64](#) geralmente disponível em seu sistema operacional. Como alternativa, você pode usar bash (defina o parâmetro bs entre 32 e 1024)

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

ou Python da seguinte forma:

```
import random
import os
import base64

# key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

Política de atualização: NOVA POLÍTICA DE ATUALIZAÇÃO COM FROTA DE COMPUTAÇÃO E NÓS DE LOGIN INTERROMPIDOS (erroneamente não adicionada na versão 3.7.0).

QueueUpdateStrategy (Opcional, **String**)

Especifica a estratégia de substituição para os parâmetros da seção [SlurmQueues](#) que têm a seguinte política de atualização:

[Política de atualização: a frota de computação deve ser interrompida ou QueueUpdateStrategy para que essa configuração seja alterada para uma atualização.](#)

O valor QueueUpdateStrategy é usado somente quando um processo de atualização do cluster é iniciado.

Valores válidos: COMPUTE_FLEET_STOP | DRAIN | TERMINATE

Valor padrão: COMPUTE_FLEET_STOP

DRAIN

Os nós em filas com valores de parâmetros alterados são definidos como DRAINING. Os nós nesse estado não aceitam novos trabalhos e os trabalhos em execução continuam sendo concluídos.

Depois que um nó se torna `idle` (DRAINED), um nó é substituído se o nó for estático e o nó será encerrado se o nó for dinâmico. Outros nós em outras filas sem valores de parâmetros alterados não são afetados.

O tempo que essa estratégia precisa para substituir todos os nós da fila por valores de parâmetros alterados depende da workload em execução.

COMPUTE_FLEET_STOP

O valor padrão do parâmetro é `QueueUpdateStrategy`. Com essa configuração, a atualização dos parâmetros na seção [SlurmQueues](#) exige que você [interrompa a frota de computação](#) antes de realizar uma atualização do cluster:

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

TERMINATE

Em filas com valores de parâmetros alterados, os trabalhos em execução são encerrados e os nós são desligados imediatamente.

Os nós estáticos são substituídos e os nós dinâmicos são encerrados.

Outros nós em outras filas sem valores de parâmetros alterados não são afetados.

[Política de atualização: essa configuração não é analisada durante uma atualização.](#)


Note

`QueueUpdateStrategy` é suportado a partir do AWS ParallelCluster versão 3.2.0.

EnableMemoryBasedScheduling (Opcional, **Boolean**)


Se `true`, o agendamento baseado em memória estiver habilitado em Slurm. Para obter mais informações, consulte [SlurmQueues](#) / [ComputeResources](#) / [SchedulableMemory](#).

O valor padrão é `false`.


 Warning

A ativação do agendamento baseado em memória afeta a maneira como o programador Slurm lida com tarefas e alocação de nós.

Para ter mais informações, consulte [Programação baseada em memória do Slurm](#).

 Note

`EnableMemoryBasedScheduling` é suportado a partir do AWS ParallelCluster versão 3.2.0.

 Note

[A partir da AWS ParallelCluster versão 3.7.0, `EnableMemoryBasedScheduling` pode ser ativado se você configurar vários tipos de instância em \[Instâncias\]\(#\).](#)

Para AWS ParallelCluster as versões 3.2.0 a 3.6. ~~x~~, não `EnableMemoryBasedScheduling` pode ser ativado se você configurar vários tipos de instância em [Instâncias](#).

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

CustomSlurmSettings (Opcional, **[Dict]**)

Define as configurações de Slurm personalizadas que se aplicam a todo o cluster.

Especifica uma lista de dicionários de configuração de pares de valores-chave Slurm a serem anexados ao final do arquivo `slurm.conf` que é gerado pelo AWS ParallelCluster .

Cada dicionário na lista aparece como uma linha separada adicionada ao arquivo de configuração Slurm. Você pode especificar parâmetros simples ou complexos.

Parâmetros simples consistem em um único par de chaves, conforme mostrado nos exemplos a seguir:

- Param1: 100
- Param2: "SubParam1,SubParam2=SubValue2"

Exemplo renderizado na configuração Slurm:

```
Param1=100
Param2=SubParam1,SubParam2=SubValue2
```

Os parâmetros de configuração complexos do Slurm consistem em vários pares de valores-chave separados por espaço, conforme mostrado nos exemplos a seguir:

- 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

Exemplo, renderizado na configuração 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

Os nós personalizados do Slurm não devem conter os padrões `-st-` ou `-dy-` em seus nomes. Esses padrões são reservados para nós gerenciados pelo AWS ParallelCluster.

Se você especificar parâmetros de configuração Slurm personalizados em `CustomSlurmSettings`, não deverá especificar parâmetros de configuração Slurm personalizados para `CustomSlurmSettingsIncludeFile`.

Você só pode especificar parâmetros de configuração Slurm que não estejam listados como negados em `CustomSlurmSettings`. Para obter mais informações sobre parâmetros de


configuração negados do Slurm, consulte [Parâmetros de configuração do Slurm listados como negados para CustomSlurmSettings](#).

AWS ParallelCluster só verifica se um parâmetro está em uma lista de negação. AWS ParallelCluster não valida a sintaxe ou a Slurm semântica do parâmetro de configuração personalizado. Você é responsável por validar seus parâmetros de configuração personalizados do Slurm. Parâmetros de configuração personalizados do Slurm inválidos podem causar falhas no daemon do Slurm que podem levar a falhas na criação e atualização do cluster.

Para obter mais informações sobre como especificar parâmetros Slurm de configuração personalizados com AWS ParallelCluster, consulte [Personalização de configuração do Slurm](#).

Para obter mais informações sobre os parâmetros de configuração Slurm, consulte [slurm.conf](#) na documentação do Slurm.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

 Note

CustomSlurmSettings é suportado a partir do AWS ParallelCluster versão 3.6.0.


CustomSlurmSettingsIncludeFile (Opcional, **String**)

Define as configurações de Slurm personalizadas que se aplicam a todo o cluster.

Especifica o arquivo Slurm personalizado que consiste em parâmetros de configuração Slurm personalizados a serem anexados ao final do arquivo `slurm.conf` que é gerado pelo AWS ParallelCluster .

Você deve incluir o caminho para o arquivo. O caminho do arquivo pode começar com `https://` ou `s3://`.

Se você especificar parâmetros de configuração Slurm personalizados para `CustomSlurmSettingsIncludeFile`, você não deverá especificar parâmetros de configuração Slurm personalizados para `CustomSlurmSettings`.

 Note

Os nós personalizados do Slurm não devem conter os padrões `-st-` ou `-dy-` em seus nomes. Esses padrões são reservados para nós gerenciados pelo AWS ParallelCluster.


Você só pode especificar parâmetros de configuração Slurm que não estejam listados como negados em `CustomSlurmSettingsIncludeFile`. Para obter mais informações sobre parâmetros de configuração negados do Slurm, consulte [Parâmetros de configuração do Slurm listados como negados para CustomSlurmSettings](#).

AWS ParallelCluster só verifica se um parâmetro está em uma lista de negação. AWS ParallelCluster não valida a sintaxe ou a Slurm semântica do parâmetro de configuração personalizado. Você é responsável por validar seus parâmetros de configuração personalizados do Slurm. Parâmetros de configuração personalizados do Slurm inválidos podem causar falhas no daemon do Slurm que podem levar a falhas na criação e atualização do cluster.

Para obter mais informações sobre como especificar parâmetros Slurm de configuração personalizados com AWS ParallelCluster, consulte [Personalização de configuração do Slurm](#).

Para obter mais informações sobre os parâmetros de configuração Slurm, consulte [slurm.conf](#) na documentação do Slurm.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

 Note

`CustomSlurmSettings` é suportado a partir do AWS ParallelCluster versão 3.6.0.

Database

(Opcional) Define as configurações para ativar a contabilidade de Slurm no cluster. Para ter mais informações, consulte [Slurmcontabilidade com AWS ParallelCluster](#).

Database:

Uri: *string*

UserName: *string*

PasswordSecretArn: *string*

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

Propriedades do **Database**

Uri (Obrigatório, **String**)

O endereço do servidor de banco de dados usado como back-end para a contabilidade Slurm. Esse URI deve ser formatado como `host:port` e não deve conter um esquema, como `mysql://`. O host pode ser um endereço IP ou um nome DNS que pode ser resolvido pelo nó principal. Se uma porta não for fornecida, AWS ParallelCluster usa a porta MySQL padrão 3306.

AWS ParallelCluster inicializa o banco de dados Slurm contábil no cluster e deve acessar o banco de dados.

O banco de dados deve estar acessível antes que o seguinte ocorra:

- Um cluster é criado.
- A contabilidade de Slurm é habilitada com uma atualização de cluster.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

UserName (Obrigatório, **String**)

A identidade usada por Slurm para se conectar ao banco de dados, gravar logs contábeis e realizar consultas. O usuário deve ter as permissões de leitura e gravação no banco de dados.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

PasswordSecretArn (Obrigatório, **String**)

O Amazon Resource Name (ARN) do AWS Secrets Manager segredo que contém a senha em texto `Username` simples. Essa senha é usada junto com a contabilidade `Username` a Slurm para autenticar no servidor de banco de dados.

Note

Ao criar um segredo usando o AWS Secrets Manager console, certifique-se de selecionar “Outro tipo de segredo”, selecionar texto simples e incluir apenas o texto da senha no segredo.

Para obter mais informações sobre como usar AWS Secrets Manager para criar um segredo, consulte [Criar um AWS Secrets Manager segredo](#)

Se o usuário tiver permissão para PasswordSecretArn isso [DescribeSecret](#), é validado. PasswordSecretArn é válido se o segredo especificado existir. Se a política do IAM do usuário não incluir DescribeSecret, PasswordSecretArn não for validada e uma mensagem de aviso será exibida. Para ter mais informações, consulte [Política básica de usuário pcluster do AWS ParallelCluster](#).

Quando você atualiza PasswordSecretArn, a frota de computação deve ser interrompida. Se o valor secreto mudar e o ARN secreto não mudar, o cluster não será atualizado automaticamente com a nova senha do banco de dados. Para atualizar o cluster para o novo valor secreto, você deve executar o comando a seguir de dentro do nó principal depois que a frota de computação for interrompida.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

Warning

Recomendamos que você altere a senha do banco de dados somente quando a frota de computação for interrompida para evitar a perda de dados contábeis.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

DatabaseName (Opcional, **String**)

Nome do banco de dados no servidor de banco de dados (definido pelo parâmetro Uri) a ser usado para o Slurm Accounting.

O nome do banco de dados pode conter letras minúsculas, números e sublinhados. O nome não pode ter mais de 64 caracteres.

Esse parâmetro é mapeado para o StorageLoc parâmetro [slurmdbd.conf](#).

Se não DatabaseName for fornecido, ParallelCluster usará o nome do cluster para definir um valor para StorageLoc.

A atualização do DatabaseName é permitida, com as seguintes considerações:

- Se um banco de dados com um nome ainda DatabaseName não existir no servidor de banco de dados, o slurmdbd o criará. Será sua responsabilidade reconfigurar o novo banco de dados

conforme necessário (por exemplo, adicionar as entidades contábeis — clusters, contas, usuários, associações, QoS etc.).

- Se um banco de dados com um nome `DatabaseName` já existir no servidor de banco de dados, o `slurmdbd` o usará para a funcionalidade Slurm Accounting.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

Note

Database é adicionado a partir da versão 3.3.0.

Dns

(Opcional) Define as configurações para Slurm que se aplicam a todo o cluster.

Dns:

`DisableManagedDns`: *boolean*

`HostedZoneId`: *string*

`UseEc2Hostnames`: *boolean*

Propriedades do Dns

DisableManagedDns (Opcional, **Boolean**)

Se `true`, as entradas de DNS do cluster não forem criadas e os nomes dos nós Slurm não puderem ser resolvidos.

Por padrão, AWS ParallelCluster cria uma zona hospedada do Route 53 na qual os nós são registrados quando iniciados. O valor padrão é `false`. Se `DisableManagedDns` estiver definido como `true`, a zona hospedada não será criada por AWS ParallelCluster.

Para saber como usar essa configuração para implantar clusters em sub-redes sem acesso à Internet, consulte [AWS ParallelCluster em uma sub-rede única sem acesso à internet](#).

Warning

É necessário um sistema de resolução de nomes para que o cluster funcione adequadamente. Se `DisableManagedDns` estiver definido como `true`, você deverá

fornecer um sistema de resolução de nomes. Para usar o DNS padrão do EC2, defina `UseEc2Hostnames` como `true`. Como alternativa, configure seu próprio resolvidor de DNS e certifique-se de que os nomes dos nós sejam registrados quando as instâncias forem iniciadas. Por exemplo, você pode fazer isso configurando [CustomActions](#) / [OnNodeStart](#).

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

HostedZoneId (Opcional, **String**)

Define uma ID de zona hospedada personalizada do Route 53 a ser usada na resolução de nomes DNS para o cluster. Quando fornecido, AWS ParallelCluster registra os nós do cluster na zona hospedada especificada e não cria uma zona hospedada gerenciada.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

UseEc2Hostnames (Opcional, **Boolean**)

Se `true`, os nós de computação do cluster são configurados com o nome de host EC2 padrão. O `NodeHostName` do Slurm também é atualizado com essas informações. O padrão é `false`.

Para saber como usar essa configuração para implantar clusters em sub-redes sem acesso à Internet, consulte [AWS ParallelCluster em uma sub-rede única sem acesso à internet](#).

Note

Esta nota não é relevante a partir do AWS ParallelCluster versão 3.3.0.

Para versões AWS ParallelCluster compatíveis anteriores à 3.3.0:

Quando `UseEc2Hostnames` definido como `true`, o arquivo de configuração do Slurm é definido com os scripts AWS ParallelCluster `prolog` e `epilog`:

- O `prolog` é executado para adicionar informações sobre nós ao `/etc/hosts` nos nós de computação quando cada trabalho é alocado.
- O `epilog` é executado para limpar o conteúdo escrito pelo `prolog`.

Para adicionar scripts personalizados `prolog` ou `epilog`, adicione-os às pastas `/opt/slurm/etc/pcluster/prolog.d/` ou `/opt/slurm/etc/pcluster/epilog.d/`, respectivamente.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Seção SharedStorage

(Opcional) As configurações de armazenamento compartilhado do cluster.

AWS ParallelCluster [suporta o uso de Amazon EBS, FSx for ONTAP e FSx paravolumes de armazenamento compartilhado OpenZFS, Amazon EFS e FSx forLustre, sistemas de arquivos de armazenamento compartilhado ou caches de arquivos.](#)

Na seção SharedStorage, você pode definir armazenamento externo ou gerenciado:

- O armazenamento externo se refere a um volume ou sistema de arquivos existente que você gerencia. AWS ParallelCluster não o cria nem o exclui.
- AWS ParallelCluster armazenamento gerenciado se refere a um volume ou sistema de arquivos AWS ParallelCluster criado e que pode ser excluído.

Para [cotas de armazenamento compartilhado](#) e obter mais informações sobre como configurar seu armazenamento compartilhado, consulte [Armazenamento compartilhado](#) em Como usar o AWS ParallelCluster.

Note

Se AWS Batch for usado como programador, o FSx for Lustre só estará disponível no nó principal do 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
- 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 políticas de atualização

- Para EBS gerenciado/externo, EFS gerenciado e FSx Lustre gerenciado, a política de atualização é [Política de atualização: para essa configuração de valores de lista, a frota de computação deve ser interrompida ou QueueUpdateStrategy deve ser configurada para agregar um novo valor; a frota de computação deve ser interrompida ao remover um valor existente.](#)
- Para EFS externos, FSx Lustre, FSx ONTAP, OpenZfs FSx e File Cache, a política de atualização é, [Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Propriedades do SharedStorage

MountDir (Obrigatório, String)

O caminho para o local em que o armazenamento compartilhado está montado.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Name (Obrigatório, String)

O nome do armazenamento compartilhado. Use esse nome ao atualizar as configurações.

Warning

Se você especificar armazenamento compartilhado AWS ParallelCluster gerenciado e alterar o valor paraName, o armazenamento compartilhado gerenciado e os dados

existentes serão excluídos e um novo armazenamento compartilhado gerenciado será criado. Alterar o valor de Name com uma atualização de cluster equivale a substituir o armazenamento compartilhado gerenciado existente por um novo. Certifique-se de fazer backup de seus dados antes de alterar o Name se precisar reter os dados do armazenamento compartilhado existente.

Política de atualização: para essa configuração de valores de lista, a frota de computação deve ser interrompida ou QueueUpdateStrategy deve ser configurada para agregar um novo valor; a frota de computação deve ser interrompida ao remover um valor existente.

StorageType (Obrigatório, String)

O tipo do armazenamento compartilhado. Os valores compatíveis são Ebs, Efs, FsxLustre, FsxOntap e FsxOpenZfs.

Para obter mais informações, consulte [FsxLustreSettings](#), [FsxOntapSettings](#) e [FsxOpenZfsSettings](#).

Note

Se você usar AWS Batch como programador, o FSx for Lustre só estará disponível no nó principal do cluster.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

EbsSettings

(Opcional) As configurações de um volume do 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
```

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do **EbsSettings**

Quando definido como `Delete`, um volume gerenciado, com seus dados, é excluído se o cluster for excluído ou se o volume for removido com uma atualização do cluster. [DeletionPolicy](#)

Para obter mais informações, consulte [Armazenamento compartilhado](#) em Como usar o AWS ParallelCluster.

VolumeType (Opcional, String)

Especifica o [tipo de volume do Amazon EBS](#). Os valores compatíveis são `gp2`, `gp3`, `io1`, `io2`, `sc1`, `st1` e `standard`. O valor padrão é `gp3`.

Para ter mais informações, consulte [Tipos de volumes do Amazon EBS](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Iops (Opcional, Integer)

Define o número de IOPS para os volumes do tipo `io1`, `io2` e `gp3`.

O valor padrão, os valores suportados e a proporção do `volume_iops` para `volume_size` variam de acordo com `VolumeType` e `Size`.

`VolumeType = io1`

Padrão Iops = 100

Valores suportados Iops = 100–64000 †

Proporção máxima de `volume_iops` para `volume_size` = 50 IOPS para cada GiB. 5000 IOPS exigem um `volume_size` de pelo menos 100 GiB.

`VolumeType = io2`

Padrão Iops = 100

Valores suportados Iops = 100–64000 (256000 para volumes do io2 Block Express) †

Proporção máxima de Iops para Size = 500 IOPS para cada GiB. 5000 IOPS exigem um Size de pelo menos 10 GiB.

VolumeType = gp3

Padrão Iops = 3000

Valores suportados Iops = 3000–16000

Proporção máxima de Iops para Size = 500 IOPS para cada GiB. 5000 IOPS exigem um Size de pelo menos 10 GiB.

† O número máximo de IOPS é garantido somente em [instâncias criadas no Nitro System](#) provisionadas com mais de 32.000 IOPS. Outras instâncias garantem até 32.000 IOPS. A menos que você [modifique o volume](#), os volumes de io1 mais antigos podem não atingir o desempenho total. io2 Os volumes do Block Express oferecem suporte a valores volume_iops de até 256000 em tipos de instância R5b. Para obter mais informações, consulte [Volumes do io2 Block Express](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Size (Opcional, Integer)

Especifica o tamanho do volume raiz em gibibytes (GiB). O valor padrão é 35.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Encrypted (Opcional, Boolean)

Especifica se o volume está criptografado. O valor padrão é true.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

KmsKeyId (Opcional, String)

Especifica uma AWS KMS chave personalizada a ser usada para criptografia. Essa configuração requer que a configuração Encrypted seja definida como true.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

SnapshotId (Opcional, String)


Especifica o ID do snapshot do Amazon EBS, caso esteja usando um snapshot como a origem do volume.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

VolumeId (Opcional, String)

Especifica o ID de volume do Amazon EBS. Quando isso é especificado para uma instância EbsSettings, somente o parâmetro MountDir também pode ser especificado.

O volume deve ser criado na mesma zona de disponibilidade do HeadNode.

 Note

Várias zonas de disponibilidade foram adicionadas na AWS ParallelCluster versão 3.4.0.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Throughput (Opcional, Integer)

O throughput em MiB/s para provisionar um volume, com um máximo de 1.000 MiB/s.

Isso é válido somente quando VolumeType é gp3. O intervalo compatível é de 125 a 1000, com um valor padrão de 125.

Política de atualização: essa configuração pode ser alterada durante uma atualização.


DeletionPolicy (Opcional, String)

Especifica se o volume deve ser retido, excluído ou capturado quando o cluster é excluído ou o volume é removido. Os valores suportados são Delete, Retain e Snapshot. O valor padrão é Delete.

Quando [DeletionPolicy](#) definido como Delete, um volume gerenciado, com seus dados, é excluído se o cluster for excluído ou se o volume for removido com uma atualização do cluster.

Para ter mais informações, consulte [Armazenamento compartilhado](#).

Política de atualização: essa configuração pode ser alterada durante uma atualização.

 Note

DeletionPolicy é suportado a partir da AWS ParallelCluster versão 3.2.0.

Raid

(Opcional) Define a configuração de um volume RAID.

Raid:

Type: *string*

NumberOfVolumes: *integer*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do **Raid**

Type (Obrigatório, String)

Define o tipo de matriz RAID. Os valores suportados são "0" (distribuído) e "1" (espelhado).

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

NumberOfVolumes (Opcional, Integer)

Define o número de volumes do Amazon EBS a usar para criar a matriz de RAID. O intervalo de valores compatível é de 2 a 5. O valor padrão (quando a configuração Raid é definida) é 2.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

EfsSettings

(Opcional) As configurações de um sistema de arquivos do Amazon EFS.

EfsSettings:

Encrypted: *boolean*

KmsKeyId: *string*

EncryptionInTransit: *boolean*

IamAuthorization: *boolean*

PerformanceMode: *string*

ThroughputMode: *string*

ProvisionedThroughput: *integer*

FileSystemId: *string*

DeletionPolicy: *string*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do `EfsSettings`

Quando [DeletionPolicy](#) definido como `Delete`, um sistema de arquivos gerenciado, com seus dados, é excluído se o cluster for excluído ou se o sistema de arquivos for removido com uma atualização do cluster.

Para obter mais informações, consulte [Armazenamento compartilhado](#) em Como usar o AWS ParallelCluster.

Encrypted (Opcional, Boolean)

Especifica se o sistema de arquivos Amazon EFS está criptografado. O valor padrão é `false`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

KmsKeyId (Opcional, String)

Especifica uma AWS KMS chave personalizada a ser usada para criptografia. Essa configuração requer que a configuração `Encrypted` seja definida como `true`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

EncryptionInTransit (Opcional, Boolean)

Se definido como `true`, os sistemas de arquivos do Amazon EFS são montados usando Transport Layer Security (TLS). Por padrão, isso é definido como `false`.

Note

Se AWS Batch for usado como agendador, `EncryptionInTransit` não é suportado.

Note

`EncryptionInTransit` é adicionado a partir do AWS ParallelCluster versão 3.4.0.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

IamAuthorization (Opcional, Boolean)

Se definido como `true`, o Amazon EFS é autenticado usando a identidade IAM do sistema. Por padrão, isso é definido como `false`.

Note

Se `IamAuthorization` estiver definido como `true`, o `EncryptionInTransit` também deverá ser definido como `true`.

Note

Se AWS Batch for usado como agendador, `IamAuthorization` não é suportado.

Note

`IamAuthorization` é adicionado a partir da AWS ParallelCluster versão 3.4.0.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

PerformanceMode (Opcional, String)

Especifica o modo de desempenho do sistema de arquivos do Amazon EFS. Os valores compatíveis são `generalPurpose` e `maxIO`. O valor padrão é `generalPurpose`. Para obter mais informações, consulte [Performance modes](#) (Modos de performance) no Guia do usuário do Amazon Elastic File System.

Recomendamos o modo de desempenho `generalPurpose` para a maioria dos sistemas de arquivos.

Os sistemas de arquivos que usam o modo de desempenho `maxIO` podem ser dimensionados para níveis superiores de throughput e operações por segundo agregadas. No entanto, há um compromisso com latências um pouco mais altas para a maioria das operações de arquivo.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

ThroughputMode (Opcional, String)

Especifica o modo de throughput do sistema de arquivos do Amazon EFS. Os valores compatíveis são `bursting` e `provisioned`. O valor padrão é `bursting`. Quando o `provisioned` é usado, `ProvisionedThroughput` deve ser especificado.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

ProvisionedThroughput (Exigido quando ThroughputMode é provisioned, Integer)

Define o throughput (em MiB/s) provisionada do sistema de arquivos Amazon EFS, medida em MiB/s. Isso corresponde ao [ProvisionedThroughputInMibps](#) parâmetro na Referência da API do Amazon EFS.

Se você usar esse parâmetro, deverá definir ThroughputMode como provisioned.

O intervalo compatível é 1-1024. Para solicitar um aumento de limite, entre em contato com o AWS Support.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

FileSystemId (Opcional, String)

Define o ID do sistema de arquivos do Amazon EFS para um sistema de arquivos existente.

Se o cluster estiver configurado para abranger várias zonas de disponibilidade, você deverá definir um destino de montagem do sistema de arquivos em cada zona de disponibilidade usada pelo cluster.

Quando isso é especificado, somente MountDir pode ser especificado. Nenhum outro EfsSettings pode ser especificado.

Se você definir essa opção, o seguinte deverá ocorrer para os sistemas de arquivos definidos:

- Sistemas de arquivos que têm um destino de montagem existente na zona de disponibilidade do cluster, com o tráfego de entrada e saída do NFS permitidos a partir de HeadNode e ComputeNodes. Várias zonas de disponibilidade são configuradas em [Scheduling//Networking](#) [SlurmQueues/SubnetIds](#).


Para garantir que o tráfego seja permitido entre o cluster e o sistema de arquivos, você pode fazer o seguinte:

- Configure os grupos de segurança do destino de montagem para permitir o tráfego de e para o CIDR ou a lista de prefixos das sub-redes do cluster.


Note

AWS ParallelCluster valida se as portas estão abertas e se o CIDR ou a lista de prefixos estão configurados. AWS ParallelCluster não valida o conteúdo do bloco CIDR ou da lista de prefixos.


- Defina grupos de segurança personalizados para nós de cluster usando [SlurmQueues / Networking / SecurityGroups](#) e [HeadNode / Networking / SecurityGroups](#). Os grupos de segurança personalizados devem ser configurados para permitir o tráfego entre o cluster e o sistema de arquivos.

 Note

Se todos os nós do cluster usarem grupos de segurança personalizados, AWS ParallelCluster somente valida se as portas estão abertas. AWS ParallelCluster não valida se a origem e o destino estão configurados corretamente.

 Warning

OneZone O EFS só é suportado se todos os nós de computação e o nó principal estiverem na mesma zona de disponibilidade. O EFS só OneZone pode ter um alvo de montagem.

 Note

Várias zonas de disponibilidade foram adicionadas na AWS ParallelCluster versão 3.4.0.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

DeletionPolicy (Opcional, String)

Especifica se o sistema de arquivos deve ser retido ou excluído quando o sistema de arquivos é removido do cluster ou o cluster é excluído. Os valores suportados são Delete e Retain. O valor padrão é Delete.

Quando definido como Delete, um sistema de arquivos gerenciado, com seus dados, é excluído se o cluster for excluído ou se o sistema de arquivos for removido com uma atualização do cluster. [DeletionPolicy](#)

Para ter mais informações, consulte [Armazenamento compartilhado](#).

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

`DeletionPolicy` é suportado a partir da AWS ParallelCluster versão 3.3.0.

FsxLustreSettings

Note

Você deve definir `FsxLustreSettings` se `FsxLustre` é especificado para [StorageType](#).

(Opcional) As configurações de um sistema de arquivos FSx para 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
```

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

Se AWS Batch for usado como programador, o FSx for Lustre só estará disponível no nó principal do cluster.

Propriedades do **FsxLustreSettings**

Quando definido como `Delete`, um sistema de arquivos gerenciado, com seus dados, é excluído se o cluster for excluído ou se o sistema de arquivos for removido com uma atualização do cluster.

[DeletionPolicy](#)

Para ter mais informações, consulte [Armazenamento compartilhado](#).

StorageCapacity (Obrigatório, Integer)

Define a capacidade de armazenamento do sistema de arquivos FSx para Lustre, em GiB. `StorageCapacity` é necessário se você estiver criando um novo sistema de arquivos. Não inclua `StorageCapacity` se `BackupId` ou `FileSystemId` estiver especificado.

- Para tipos de implantação `SCRATCH_2`, `PERSISTENT_1` e `PERSISTENT_2`, os valores válidos são de 1200 GiB, 2400 GiB e incrementos de 2400 GiB.
- Para o tipo de implantação `SCRATCH_1`, os valores válidos são de 1200 GiB, 2400 GiB e incrementos de 3600 GiB.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

DeploymentType (Opcional, String)

Especifica o tipo de implantação do sistema de arquivos FSx para Lustre. Os valores compatíveis são `SCRATCH_1`, `SCRATCH_2`, `PERSISTENT_1` e `PERSISTENT_2`. O valor padrão é `SCRATCH_2`.

Escolha os tipos de implantação `SCRATCH_1` e `SCRATCH_2` quando precisar de armazenamento temporário e processamento de dados de prazo mais curto. O tipo de implantação `SCRATCH_2` fornece criptografia de dados em trânsito e capacidade de throughput de intermitência mais alta que o `SCRATCH_1`.

Escolha o tipo de implantação `PERSISTENT_1` para armazenamento de longo prazo e para workloads focadas no throughput que não sejam sensíveis à latência. O `PERSISTENT_1` oferece suporte à criptografia de dados em trânsito. Ele está disponível em todos os Regiões da AWS lugares onde o FSx for Lustre está disponível.

Escolha o tipo de implantação `PERSISTENT_2` para armazenamento de prazo mais longo e para workloads sensíveis à latência que exigem os níveis mais altos de IOPS e throughput. `PERSISTENT_2` é compatível com o armazenamento SSD e oferece maior `PerUnitStorageThroughput` (até 1000 MB/s/TiB). `PERSISTENT_2` está disponível em um número limitado de Regiões da AWS. Para obter mais informações sobre os tipos de

implantação e a lista de Regiões da AWS onde PERSISTENT_2 está disponível, consulte [Opções de implantação do sistema de arquivos para FSx for Lustre no Guia](#) do usuário do Amazon FSx for Lustre.

A criptografia de dados em trânsito é habilitada automaticamente quando você acessa um sistema de arquivos de tipo de implantação SCRATCH_2, PERSISTENT_1 ou PERSISTENT_2 de instâncias do Amazon EC2 que são compatíveis com [esse recurso](#).

A criptografia de dados em trânsito dos tipos de implantação SCRATCH_2, PERSISTENT_1 e PERSISTENT_2 é compatível quando acessada de tipos de instância compatíveis em Regiões da AWS compatíveis. Para obter mais informações, consulte [Criptografia de dados em trânsito](#) no Guia do usuário do Amazon FSx para Lustre.

Note

Suporte para o tipo de implantação PERSISTENT_2 foi adicionado com o AWS ParallelCluster versão 3.2.0.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

ImportedFileChunkSize (Opcional, Integer)

Para arquivos que são importados de um repositório de dados, esse valor determina a contagem de stripes e a quantidade máxima de dados para cada arquivo (em MiB) armazenados em um único disco físico. O número máximo de discos nos quais um único arquivo pode ser distribuído é limitado pelo número total de discos que compõem o sistema de arquivos.

O tamanho de bloco padrão é 1.024 MiB (1 GiB) e pode chegar até 512.000 MiB (500 GiB). Os objetos do Amazon S3 têm um tamanho máximo de 5 TB.

Note

Esse parâmetro não é compatível com sistemas de arquivos que usam o tipo de implantação PERSISTENT_2. Para obter instruções sobre como configurar associações de repositórios de dados, consulte [Como vincular seu sistema de arquivos a um bucket do S3](#) no Guia do usuário do Amazon FSx para Lustre.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

DataCompressionType (Opcional, String)

A configuração de compactação de dados do sistema de arquivos do FSx para Lustre. O valor compatível é LZ4. LZ4 indica que a compactação de dados está ativada com o algoritmo LZ4. Quando DataCompressionType não é especificado, a compactação de dados é desativada quando o sistema de arquivos é criado.

Para obter mais informações, consulte [Compactação de dados do Lustre](#).

Política de atualização: essa configuração pode ser alterada durante uma atualização.

ExportPath (Opcional, String)

O caminho no Amazon S3 em que a raiz do sistema de arquivos FSx para Lustre é exportada. Essa configuração apenas tem suporte quando o parâmetro ImportPath é especificado. O caminho deve usar o mesmo bucket do Amazon S3 conforme especificado em ImportPath. Você pode fornecer um prefixo opcional para o qual os dados novos e alterados devem ser exportados do sistema de arquivos FSx para Lustre. Se um valor ExportPath não for fornecido, o FSx para Lustre definirá um caminho de exportação padrão, `s3://import-bucket/FSxLustre[creation-timestamp]`. O timestamp é no formato UTC, por exemplo `s3://import-bucket/FSxLustre20181105T222312Z`.

O bucket de exportação do Amazon S3 deve ser o mesmo que o bucket de importação especificado por ImportPath. Se especificar apenas um nome de bucket, como `s3://import-bucket`, você obterá um mapeamento de 1:1 de objetos do sistema de arquivos para os objetos do bucket do Amazon S3. Esse mapeamento indica que os dados de entrada no Amazon S3 são substituídos na exportação. Se você fornecer um prefixo personalizado no caminho de exportação, como `s3://import-bucket/[custom-optional-prefix]`, o FSx para Lustre exportará o conteúdo de seu sistema de arquivos para esse prefixo de exportação no bucket do Amazon S3.

Note

Esse parâmetro não é compatível com sistemas de arquivos que usam o tipo de implantação PERSISTENT_2. Configure associações de repositórios de dados conforme descrito em [Como vincular seu sistema de arquivos a um bucket do S3](#) no Guia do usuário do Amazon FSx para Lustre.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

ImportPath (Opcional, String)

O caminho para o bucket do Amazon S3 (incluindo o prefixo opcional) que você está usando como o repositório de dados do sistema de arquivos FSx para Lustre. A raiz do sistema de arquivos do FSx para Lustre será mapeada para a raiz do bucket do Amazon S3 que você selecionar. Um exemplo é `s3://import-bucket/optional-prefix`. Se você especificar um prefixo após o nome do bucket do Amazon S3, apenas chaves de objeto com esse prefixo serão carregadas no sistema de arquivos.

Note

Esse parâmetro não é compatível com sistemas de arquivos que usam o tipo de implantação `PERSISTENT_2`. Configure associações de repositórios de dados conforme descrito em [Como vincular seu sistema de arquivos a um bucket do S3](#) no Guia do usuário do Amazon FSx para Lustre.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

WeeklyMaintenanceStartTime (Opcional, String)

O horário de início preferido para realizar a manutenção semanal. Fica em formato "d:HH:MM" do fuso horário UTC+0. Para esse formato, d é o número do dia da semana de 1 a 7, começando com a segunda-feira e terminando no domingo. As aspas são obrigatórias para esse campo.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

AutomaticBackupRetentionDays (Opcional, Integer)

O número de dias em que os backups automáticos serão retidos. Definir esta opção como 0 desabilita os backups automáticos. O intervalo compatível é 0-90. O padrão é 0. Essa configuração só é válida para uso com os tipos de implantação `PERSISTENT_1` e `PERSISTENT_2`. Para obter mais informações, consulte [Trabalhar com backups](#) no Guia do usuário do Amazon FSx para Lustre.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

CopyTagsToBackups (Opcional, Boolean)

Se `true`, copie as tags do sistema de arquivos FSx para Lustre para os backups. Esse valor padrão é `false`. Se estiver definido como `true`, todas as tags serão copiadas em todos os

backups automáticos e iniciados pelo usuário quando nenhuma tag for especificada. Se esse valor for `true` e você especificar uma ou mais tags, apenas as que forem especificadas serão copiadas nos backups. Se você especificar uma ou mais tags ao criar um backup iniciado pelo usuário, nenhuma tag será copiada do sistema de arquivos, independentemente desse valor. Essa configuração só é válida para uso com os tipos de implantação `PERSISTENT_1` e `PERSISTENT_2`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

`DailyAutomaticBackupStartTime` (Opcional, `String`)

Um horário diário recorrente, no formato `HH:MM`. `HH` é a hora do dia preenchida com zeros (00-23). `MM` é o minuto da hora preenchido com zeros (00-59). Por exemplo, `05:00` especifica 5h da manhã diariamente. Essa configuração só é válida para uso com os tipos de implantação `PERSISTENT_1` e `PERSISTENT_2`.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

`PerUnitStorageThroughput` (Obrigatório para os tipos de implantação **`PERSISTENT_1`** e **`PERSISTENT_2`**, `Integer`)

Descreve a quantidade de throughput de leitura e gravação para cada 1 tebibyte de armazenamento, em MB/s/TiB. A capacidade de throughput do sistema de arquivos é calculada multiplicando-se a capacidade de armazenamento do sistema de arquivos (TiB) pela `PerUnitStorageThroughput` (MB/s/TiB). Para um sistema de arquivos de 2,4 TiB, o provisionamento de 50 MB/s/TiB `PerUnitStorageThroughput` gera 120 MB/s de throughput do sistema de arquivos. Você paga pela quantidade de throughput que provisiona. Isso corresponde à [PerUnitStorageThroughput](#) propriedade.

Valores válidos:

Armazenamento `PERSISTENT_1` SSD: 50, 100, 200 MB/s/TiB.

Armazenamento `PERSISTENT_1` HDD: 12, 40 MB/s/TiB.

Armazenamento `PERSISTENT_2` SSD: 125, 250, 500, 1000 MB/s/TiB.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

`BackupId` (Opcional, `String`)

Especifica o ID do backup a ser usado para restaurar o sistema de arquivos FSx para Lustre a partir de um backup existente. Quando a configuração `BackupId` é

especificada, as configurações `AutoImportPolicy`, `DeploymentType`, `ExportPath`, `KmsKeyId`, `ImportPath`, `ImportedFileChunkSize`, `StorageCapacity`, e `PerUnitStorageThroughput` não devem ser especificadas. Essas configurações são lidas do backup. Além disso, as configurações `AutoImportPolicy`, `ExportPath`, `ImportPath` e `ImportedFileChunkSize` não devem ser especificados. Isso corresponde à [BackupId](#) propriedade.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

`KmsKeyId` (Opcional, `String`)

A ID da chave AWS Key Management Service (AWS KMS) usada para criptografar os dados do sistema de arquivos FSx for Lustre para sistemas de arquivos FSx for Lustre persistentes em repouso. Se não for especificada, a chave gerenciada do FSx para Lustre será usada. Os sistemas de arquivos FSx para Lustre `SCRATCH_1` e `SCRATCH_2` são sempre criptografados em repouso usando chaves gerenciadas do FSx para Lustre. Para obter mais informações, consulte [Criptografia](#) na Referência da API do AWS Key Management Service .

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

`FileSystemId` (Opcional, `String`)

Especifica a ID de um sistema de arquivos FSx para Lustre existente.

Se essa opção for especificada, somente as configurações `MountDir` e `FileSystemId` do `FsxLustreSettings` serão usadas. Todas as outras configurações no `FsxLustreSettings` são ignoradas.

Note


Se o AWS Batch agendador for usado, o FSx for Lustre estará disponível apenas no nó principal.

Note

Se estiver usando um sistema de arquivos existente, ele deve ser associado a um grupo de segurança que permite o tráfego de entrada e saída do TCP pelas portas 988, 1021, 1022, e 1023.


Confirme que o tráfego seja permitido entre o cluster e o sistema de arquivos, fazendo o seguinte:

- Configure os grupos de segurança do sistema de arquivos para permitir o tráfego de e para o CIDR ou a lista de prefixos das sub-redes do cluster.

 Note

AWS ParallelCluster valida se as portas estão abertas e se o CIDR ou a lista de prefixos estão configurados. AWS ParallelCluster não valida o conteúdo do bloco CIDR ou da lista de prefixos.

- Defina grupos de segurança personalizados para nós de cluster usando [SlurmQueues / Networking / SecurityGroups](#) e [HeadNode / Networking / SecurityGroups](#). Os grupos de segurança personalizados devem ser configurados para permitir o tráfego entre o cluster e o sistema de arquivos.

 Note

Se todos os nós do cluster usarem grupos de segurança personalizados, AWS ParallelCluster somente valida se as portas estão abertas. AWS ParallelCluster não valida se a origem e o destino estão configurados corretamente.


Política de atualização: se essa configuração for alterada, a atualização não será permitida.

AutoImportPolicy (Opcional, String)

Quando você cria o sistema de arquivos FSx para Lustre, seus objetos existentes no Amazon S3 aparecem como listagens de arquivos e diretórios. Use essa propriedade para escolher como o FSx para Lustre mantém suas listagens de arquivos e diretórios atualizadas à medida que você adiciona ou modifica objetos no bucket do Amazon S3 vinculado. A AutoImportPolicy pode ter os seguintes valores:

- NEW - A importação automática está ativada. O FSx para Lustre importa automaticamente listagens de diretórios de quaisquer novos objetos adicionados ao bucket do Amazon S3 vinculado que não existam atualmente no sistema de arquivos FSx para Lustre.
- NEW_CHANGED - A importação automática está ativada. O FSx para Lustre importa automaticamente listagens de arquivos e diretórios de quaisquer novos objetos adicionados ao bucket do Amazon S3 e quaisquer objetos existentes que sejam alterados no bucket do Amazon S3 depois que você escolher essa opção.


- `NEW_CHANGED_DELETED` - A importação automática está ativada. O FSx para Lustre importa automaticamente listagens de arquivos e diretórios de quaisquer novos objetos adicionados ao bucket do Amazon S3, além de quaisquer objetos existentes que sejam alterados no bucket do Amazon S3 ou que foram excluídos do bucket do Amazon S3 após você escolher essa opção.

 Note

O suporte para `NEW_CHANGED_DELETED` foi adicionado ao AWS ParallelCluster versão 3.1.1.

Se `AutoImportPolicy` não for especificado, a importação automática está desativada. O FSx para Lustre atualiza somente as listagens de arquivos e diretórios do bucket do Amazon S3 vinculado quando o sistema de arquivos é criado. O FSx para Lustre não atualiza listagens de arquivos e diretórios para objetos novos ou alterados depois da escolha dessa opção.

Para obter mais informações, consulte [Importar atualizações do bucket do S3 automaticamente](#) no Guia do usuário do Amazon FSx para Lustre.

 Note

Esse parâmetro não é compatível com sistemas de arquivos que usam o tipo de implantação `PERSISTENT_2`. Para obter instruções sobre como configurar associações de repositórios de dados, consulte [Como vincular seu sistema de arquivos a um bucket do S3](#) no Guia do usuário do Amazon FSx para Lustre.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

`DriveCacheType` (Opcional, String)

Especifica que o sistema de arquivos tem um cache de unidade SSD. Isso só pode ser definido se a configuração `StorageType` estiver definida como `HDD`, e a configuração `DeploymentType` estiver definida como `PERSISTENT_1`. Isso corresponde à `DriveCacheType` propriedade. Para obter mais informações, consulte [Opções de implantação do FSx para Lustre](#) no Guia do usuário do Amazon FSx para Lustre.

O único valor válido é `READ`. Para desabilitar o cache da unidade SSD, não especifique a configuração `DriveCacheType`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

StorageType (Opcional, String)

Define o tipo de armazenamento para o sistema de arquivos FSx para Lustre que você está criando. Os valores válidos são SSD e HDD.

- Defina como SSD para usar o armazenamento da unidade de estado sólido.
- Defina como HDD para usar o armazenamento da unidade de disco rígido. HDD é compatível com os tipos de implantação PERSISTENT.

O valor padrão é SSD. Para obter mais informações, consulte [Opções de tipos de armazenamento](#) no Guia do usuário do Amazon FSx para Windows e [Várias opções de armazenamento](#) no Guia do usuário do Amazon FSx para Lustre.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

DeletionPolicy (Opcional, String)

Especifica se o sistema de arquivos deve ser retido ou excluído quando o sistema de arquivos é removido do cluster ou o cluster é excluído. Os valores suportados são Delete e Retain. O valor padrão é Delete.

Quando definido como Delete, um sistema de arquivos gerenciado, com seus dados, é excluído se o cluster for excluído ou se o sistema de arquivos for removido com uma atualização do cluster. [DeletionPolicy](#)

Para ter mais informações, consulte [Armazenamento compartilhado](#).

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

DeletionPolicy é suportado a partir da AWS ParallelCluster versão 3.3.0.

DataRepositoryAssociations (Opcional, String)

Lista de DRAs (até 8 por sistema de arquivos)

Cada associação de repositório de dados deve ter um diretório exclusivo no sistema de arquivos Amazon FSx e um bucket do S3 ou um prefixo exclusivo associado a ele.

Você não pode usar [ExportPath](#) [ImportPath](#) [FsxLustreSettings](#) ao mesmo tempo que usa DRAs.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Name (Obrigatório, String)

O nome do DRA. Use esse nome ao atualizar as configurações.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

BatchImportMetaDataOnCreate (Opcional, Boolean)

Um sinalizador booleano indicando se uma tarefa de importação do repositório de dados para importar metadados deve ser executada após a criação da associação do repositório de dados. A tarefa será executada se esse sinalizador estiver definido como `true`.

Valor padrão: `false`

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

DataRepositoryPath (Obrigatório, String)

O caminho para o repositório de dados do Amazon S3 que será vinculado sistema de arquivos. O caminho pode ser um bucket ou prefixo do S3 no formato `s3://myBucket/myPrefix/`. Esse caminho especifica de onde os arquivos do repositório de dados do S3 serão importados ou para onde serão exportados.

Não pode se sobrepor a outros DRAs

Padrão: `^[^\u0000\u0085\u2028\u2029\r\n]{3,4357}$`

Mínimo: 3

Máximo: 4357

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)


FilesystemPath (Obrigatório, String)

Um caminho no sistema de arquivos Amazon FSx para Lustre que aponta para um diretório de alto nível (como `/ns1/`) ou subdiretório (como `/ns1/subdir/`) que será mapeado diretamente com `DataRepositoryPath`. A barra inicial no nome é obrigatória. Duas associações de repositórios de dados não podem ter caminhos de sistema de arquivos sobrepostos. Por

exemplo, se um repositório de dados estiver associado ao caminho do sistema de arquivos / ns1/, você não poderá vincular outro repositório de dados ao caminho do sistema de arquivos / ns1/ns2.

Esse caminho especifica de onde os arquivos do sistema de arquivos serão exportados ou para onde eles serão importados. Esse diretório do sistema de arquivos só pode ser vinculado a um bucket do Amazon S3, e nenhum outro bucket do S3 pode ser vinculado ao diretório.

Não pode se sobrepor a outros DRAs

 Note

Se você especificar somente uma barra (/) como o caminho do sistema de arquivos, poderá vincular somente um repositório de dados ao sistema de arquivos. Você só pode especificar "/" como o caminho do sistema de arquivos para o primeiro repositório de dados associado a um sistema de arquivos.

Padrão: `^[^\u0000\u0085\u2028\u2029\r\n]{1,4096}$`

Mínimo: 1

Máximo: 4096

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

ImportedFileChunkSize (Opcional, Integer)

Para arquivos importados de um repositório de dados, esse valor determina a contagem de stripes e a quantidade máxima de dados por arquivo (em MiB) armazenados em um único disco físico. O número máximo de discos nos quais um único arquivo pode ser distribuído é limitado pelo número total de discos que compõem o sistema de arquivos ou cache.

O tamanho de bloco padrão é 1.024 MiB (1 GiB) e pode chegar até 512.000 MiB (500 GiB). Os objetos do Amazon S3 têm um tamanho máximo de 5 TB.

Mínimo: 1

Máximo: 4096

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

AutoExportPolicy (Opcional, Array of strings)

A lista pode conter um ou mais dos seguintes valores:

- **NEW:** arquivos e diretórios novos são exportados automaticamente para o repositório de dados à medida que são adicionados ao sistema de arquivos.
- **CHANGED:** as alterações em arquivos e diretórios no sistema de arquivos são exportadas automaticamente para o repositório de dados.
- **DELETED:** arquivos e diretórios novos são excluídos automaticamente no repositório de dados à medida que são excluídos do sistema de arquivos.

É possível definir qualquer combinação de tipos de eventos para seu AutoExportPolicy.

Máximo: 3

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

AutoImportPolicy (Opcional, Array of strings)

A lista pode conter um ou mais dos seguintes valores:

- **NEW:** o Amazon FSx importa automaticamente metadados de arquivos adicionados ao bucket do S3 vinculado que não existem atualmente no sistema de arquivos FSx.
- **CHANGED:** o Amazon FSx atualiza automaticamente os metadados do arquivo e invalida o conteúdo do arquivo existente no sistema de arquivos à medida que os arquivos são alterados no repositório de dados.
- **DELETED:** o Amazon FSx exclui automaticamente os arquivos no sistema de arquivos à medida que os arquivos correspondentes são excluídos no repositório de dados.

É possível definir qualquer combinação de tipos de eventos para seu AutoImportPolicy.

Máximo: 3

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

FsxOntapSettings

Note

Você deve definir FsxOntapSettings se FsxOntap é especificado para [StorageType](#).

(Opcional) As configurações de um sistema de arquivos FSx para ONTAP.

FsxOntapSettings:
VolumeId: *string*

Propriedades do **FsxOntapSettings**

VolumeId (Obrigatório, String)

Especifica a ID de volume de um sistema de arquivos FSx para ONTAP existente.

Note

- Se um AWS Batch agendador for usado, o FSx for ONTAP estará disponível somente no nó principal.
- Se o tipo de implantação do FSx para ONTAP for Multi-AZ, certifique-se de que a tabela de rotas da sub-rede do nó principal esteja configurada corretamente.
- Support for FSx for ONTAP foi adicionado na AWS ParallelCluster versão 3.2.0.
- Se estiver usando um sistema de arquivos existente, ele deve ser associado a um grupo de segurança que permite o tráfego de entrada e saída do TCP e UDP pelas portas 111, 635, 2049 e 4046.

Confirme que o tráfego seja permitido entre o cluster e o sistema de arquivos, fazendo as ações a seguir:

- Configure os grupos de segurança do sistema de arquivos para permitir o tráfego de e para o CIDR ou a lista de prefixos das sub-redes do cluster.

Note

AWS ParallelCluster valida se as portas estão abertas e se o CIDR ou a lista de prefixos estão configurados. AWS ParallelCluster não valida o conteúdo do bloco CIDR ou da lista de prefixos.

- Defina grupos de segurança personalizados para nós de cluster usando [SlurmQueues](#) / [Networking](#) / [SecurityGroups](#) e [HeadNode](#) / [Networking](#) / [SecurityGroups](#). Os grupos

de segurança personalizados devem ser configurados para permitir o tráfego entre o cluster e o sistema de arquivos.

Note

Se todos os nós do cluster usarem grupos de segurança personalizados, AWS ParallelCluster somente valida se as portas estão abertas. AWS ParallelCluster não valida se a origem e o destino estão configurados corretamente.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

FsxOpenZfsSettings

Note

Você deve definir `FsxOpenZfsSettings` se `FsxOpenZfs` é especificado para [StorageType](#).

(Opcional) As configurações de um sistema de arquivos FSx para Open ZFS.

`FsxOpenZfsSettings`:

`VolumeId`: *string*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do FsxOpenZfsSettings

`VolumeId` (Obrigatório, `String`)

Especifica a ID de volume de um sistema de arquivos FSx para Open ZFS existente.

Note

- Se um AWS Batch agendador for usado, o FSx for OpenZFS estará disponível somente no nó principal.
- Support for FSx for OpenZFS foi adicionado na versão 3.2.0. AWS ParallelCluster

- Se estiver usando um sistema de arquivos existente, ele deve ser associado a um grupo de segurança que permite o tráfego de entrada e saída do TCP e UDP pelas portas 111, 2049, 20001, 20002 e 20003.

Confirme que o tráfego seja permitido entre o cluster e o sistema de arquivos, fazendo o seguinte:

- Configure os grupos de segurança do sistema de arquivos para permitir o tráfego de e para o CIDR ou a lista de prefixos das sub-redes do cluster.

Note

AWS ParallelCluster valida se as portas estão abertas e se o CIDR ou a lista de prefixos estão configurados. AWS ParallelCluster não valida o conteúdo do bloco CIDR ou da lista de prefixos.

- Defina grupos de segurança personalizados para nós de cluster usando [SlurmQueues / Networking / SecurityGroups](#) e [HeadNode / Networking / SecurityGroups](#). Os grupos de segurança personalizados devem ser configurados para permitir o tráfego entre o cluster e o sistema de arquivos.

Note

Se todos os nós do cluster usarem grupos de segurança personalizados, AWS ParallelCluster somente valida se as portas estão abertas. AWS ParallelCluster não valida se a origem e o destino estão configurados corretamente.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

FileCacheSettings

Note

Você deve definir FileCacheSettings se FileCache é especificado para [StorageType](#).

(Opcional) As configurações de um cache de arquivos.

FileCacheSettings:**FileCacheId:** *string*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do FileCacheSettings**FileCacheId** (Obrigatório, String)

Especifica a ID do cache de arquivos de um cache de arquivos existente.

Note

- O cache de arquivos não oferece suporte a AWS Batch agendadores.
- Support for File Cache foi adicionado na AWS ParallelCluster versão 3.7.0.
- Se estiver usando um sistema de arquivos existente, ele deve ser associado a um grupo de segurança que permite o tráfego de entrada e saída do TCP pela porta 988.

Confirme que o tráfego seja permitido entre o cluster e o sistema de arquivos, fazendo o seguinte:

- Configure os grupos de segurança do cache de arquivos para permitir o tráfego de e para o CIDR ou a lista de prefixos das sub-redes do cluster.

Note

AWS ParallelCluster valida se as portas estão abertas e se o CIDR ou a lista de prefixos estão configurados. AWS ParallelCluster não valida o conteúdo do bloco CIDR ou da lista de prefixos.

- Defina grupos de segurança personalizados para nós de cluster usando [SlurmQueues / Networking](#) / [SecurityGroups](#) e [HeadNode / Networking](#) / [SecurityGroups](#). Os grupos de segurança personalizados devem ser configurados para permitir o tráfego entre o cluster e o sistema de arquivos.

Note

Se todos os nós do cluster usarem grupos de segurança personalizados, AWS ParallelCluster somente valida se as portas estão abertas. AWS ParallelCluster não valida se a origem e o destino estão configurados corretamente.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Seção **Iam**

(Opcional) Especifica as propriedades do IAM para o cluster.

Iam:

Roles:

LambdaFunctionsRole: *string*

PermissionsBoundary: *string*

ResourcePrefix: *string*

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Propriedades do **Iam**

PermissionsBoundary (Opcional, String)

O ARN da política do IAM usado como limite de permissões para todos os perfis do IAM criados pelo AWS ParallelCluster. Para obter mais informações, consulte [Limites de permissões para entidades do IAM](#) no Guia do usuário do IAM. O formato é `arn:${Partition}:iam::${Account}:policy/${PolicyName}`.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Roles (Opcional)

Especifica as configurações dos perfis do IAM usados pelo cluster.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

LambdaFunctionsRole (Opcional, String)

O ARN da função do IAM a ser usada para AWS Lambda. Isso substitui a função padrão associada a todas as funções do Lambda que apoiam recursos personalizados. AWS

CloudFormation O Lambda precisa ser configurado como entidade principal autorizada a assumir a função. Isso não substituirá a função das funções Lambda usadas para. AWS Batch O formato é `arn:${Partition}:iam::${Account}:role/${RoleName}`.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

ResourcePrefix (Opcional)

Especifica um caminho ou prefixo de nome para recursos do IAM criados por. AWS ParallelCluster

O prefixo do recurso deve seguir as [regras de nomenclatura especificadas pelo IAM:](#)

- Um nome pode conter até 30 caracteres.
- Um nome só pode ser uma string sem caracteres de barra (/).
- Um caminho pode ter até 512 caracteres.
- Um caminho deve começar e terminar com uma barra (/). Ele pode conter várias barras (/) entre as barras inicial e final (/).
- Você pode combinar o caminho e o nome /path/name.

Especifique um nome.

```
Iam:  
ResourcePrefix: my-prefix
```

Especifique um caminho.

```
Iam:  
ResourcePrefix: /org/dept/team/project/user/
```

Especifique um caminho e um nome.

```
Iam:  
ResourcePrefix: /org/dept/team/project/user/my-prefix
```

Se você especificar /my-prefix, retornará um erro.

```
Iam:
```

```
ResourcePrefix: /my-prefix
```

Um erro de configuração é retornado. Um caminho deve ter duas /s. Um prefixo por si só não pode ter /s.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Seção **LoginNodes**

Note

Support LoginNodes for adicionado na AWS ParallelCluster versão 3.7.0.

(Opcional) Especifica a configuração do grupo de nós de login.

LoginNodes:

Pools:

- Name: *string*
- Count: *integer*
- InstanceType: *string*
- GracetimePeriod: *integer*
- Image:
 - CustomAmi: *string*
- Ssh:
 - KeyName: *string*
- Networking:
 - SubnetIds:
 - *string*
 - SecurityGroups:
 - *string*
 - AdditionalSecurityGroups:
 - *string*
- Iam:
 - InstanceRole: *string*
 - InstanceProfile: *string*
 - AdditionalIamPolicies:
 - Policy: *string*

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

Propriedades do **LoginNodes**

Propriedades do **Pools**

Define grupos de nós de login que têm a mesma configuração de recursos. Somente um único grupo filho pode ser especificado.

Pools:

- Name: *string*
- Count: *integer*
- InstanceType: *string*
- GracetimePeriod: *integer*
- Image:
 - CustomAmi: *string*
- Ssh:
 - KeyName: *string*
- Networking:
 - SubnetIds:
 - *string*
 - SecurityGroups:
 - *string*
 - AdditionalSecurityGroups:
 - *string*
- Iam:
 - InstanceRole: *string*
 - InstanceProfile: *string*
 - AdditionalIamPolicies:
 - Policy: *string*

Name (Obrigatório String)

Especifica o nome do grupo LoginNodes. Isso é usado para marcar os recursos LoginNodes.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Count (Obrigatório Integer)

Especifica o número de nós de login que devem ser mantidos ativos.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

InstanceType (Obrigatório String)

Especifica o tipo de instância do Amazon EC2 que é usado para o nó de login. A arquitetura do tipo de instância deve ser a mesma usada para a configuração InstanceType do Slurm.

Política de atualização: essa configuração pode ser alterada se o grupo de nós de login for interrompido.

GracetimePeriod (Integer Opcional)

Especifica o tempo mínimo em minutos decorrido entre a notificação ao usuário conectado de que um nó de login deve ser desativado e o evento de parada real. Os valores válidos para GracetimePeriod são de 3 a 120 minutos. O padrão é 60 minutos.

Note

O evento acionador envolve interações entre vários AWS serviços. Às vezes, a latência da rede e a propagação das informações podem levar algum tempo, então o período de carência pode demorar mais do que o esperado devido a atrasos internos nos serviços. AWS

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Image (Opcional)

Define a configuração da imagem para os nós de login.

Image:

CustomAmi: *String*

CustomAmi (String Opcional)

Especifica a AMI personalizada usada para provisionar os nós de login. Se não for especificado, o valor padrão é o especificado no [Seção HeadNode](#).

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Ssh (Opcional)

Define a configuração da ssh para os nós de login.

Ssh:

KeyName: *string*

KeyName (String Opcional)

Especifica a chave ssh usada para fazer login nos nós de login. Se não for especificado, o valor padrão é o especificado no [Seção HeadNode](#).

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Networking (Obrigatório)

Networking:

SubnetIds:

- *string*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

SubnetIds (Obrigatório [String])

O ID da sub-rede existente na qual você provisiona o grupo de nós de login. Você pode definir apenas uma sub-rede.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

SecurityGroups ([String] Opcional)

Uma lista de grupos de segurança para usar no grupo de nós de login. Se nenhum grupo de segurança for especificado, AWS ParallelCluster cria grupos de segurança para você.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

AdditionalSecurityGroups ([String] Opcional)

Uma lista adicional de grupos de segurança para usar no grupo de nós de login.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Iam (Opcional)

Especifica uma função de instância ou um perfil de instância a ser usado nos nós de login para substituir a função de instância padrão ou o perfil de instância do cluster.

Iam:

```
InstanceRole: string  
InstanceProfile: string  
AdditionalIamPolicies:  
- Policy: string
```

InstanceProfile (String Opcional)

Especifica um perfil de instância para substituir o perfil padrão da instância do nó de login. Não é possível especificar ambos InstanceProfile e InstanceRole. O formato é `arn:Partition:iam::Account:instance-profile/InstanceProfileName`. Se isso for especificado, as configurações InstanceRole e AdditionalIamPolicies não poderão ser especificadas.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

InstanceRole (String Opcional)

Especifica uma função de instância para substituir o a função padrão da instância do nó de login. Não é possível especificar ambos InstanceProfile e InstanceRole. O formato é `arn:Partition:iam::Account:role/RoleName`. Se isso for especificado, as configurações S3Access e AdditionalIamPolicies não poderão ser especificadas. Se isso for especificado, as configurações InstanceProfile e AdditionalIamPolicies não poderão ser especificadas.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

AdditionalIamPolicies (Opcional)

```
AdditionalIamPolicies:  
- Policy: string
```

Um nome do recurso da Amazon (ARN) da política do IAM.

Especifica uma lista de nomes de recursos da Amazon (ARNs) de políticas do IAM para o Amazon EC2. Essa lista é anexada à função raiz usada para o nó de login, além das permissões exigidas pelo AWS ParallelCluster.

O nome da política do IAM e seu ARN são diferentes. Os nomes não podem ser usados.

Se isso for especificado, as configurações InstanceProfile e InstanceRole não poderão ser especificadas. Recomendamos que você use AdditionalIamPolicies porque

`AdditionalIamPolicies` são adicionados às permissões necessárias e `InstanceRole` devem incluir todas as permissões necessárias. AWS ParallelCluster As permissões necessárias muitas vezes são alteradas entre as versões à medida que recursos são adicionados.

Não há valor padrão.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Policy (Obrigatório [String])

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Seção **Monitoring**

(Opcional) Especifica as configurações de monitoramento do cluster.

Monitoring:

Logs:

CloudWatch:

Enabled: *boolean*

RetentionInDays: *integer*

DeletionPolicy: *string*

Rotation:

Enabled: *boolean*

Dashboards:

CloudWatch:

Enabled: *boolean*

DetailedMonitoring: *boolean*

Alarms:

Enabled: *boolean*

Política de atualização: essa configuração não é analisada durante uma atualização.

Propriedades do **Monitoring**

Logs (Opcional)

As configurações de log do cluster.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

CloudWatch (Opcional)

As configurações de CloudWatch registros para o cluster.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Enabled (Obrigatório, Boolean)

Se `true`, os registros do cluster são transmitidos para o CloudWatch Logs. O valor padrão é `true`.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

RetentionInDays (Opcional, Integer)

O número de dias para reter os eventos de registro em CloudWatch Registros. O valor padrão é 180. Os valores suportados são 0, 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827 e 3653. Um valor de 0 usará a configuração padrão de retenção de CloudWatch registros, ou seja, nunca expirará.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

DeletionPolicy (Opcional, String)

Indica se os eventos de log devem ser excluídos em CloudWatch Logs quando o cluster é excluído. Os valores possíveis são `Delete` e `Retain`. O valor padrão é `Retain`.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Rotation (Opcional)

As configurações de alternância de log do cluster.

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Enabled (Obrigatório, Boolean)

Se `true`, a alternância de log está habilitada. O padrão é `true`. Quando um arquivo de log AWS ParallelCluster configurado atinge um determinado tamanho, ele é rotacionado e um único backup é mantido. Para ter mais informações, consulte [Rotação de log configurado do AWS ParallelCluster](#).

[Política de atualização: se essa configuração for alterada, a atualização não será permitida.](#)

Dashboards (Opcional)

As configurações de painel do cluster.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

CloudWatch (Opcional)

As configurações CloudWatch do painel para o cluster.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Enabled (Obrigatório, Boolean)

Set `true`, o CloudWatch painel estiver ativado. O valor padrão é `true`.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

DetailedMonitoring (Opcional, Boolean)

Se definido como `true`, o monitoramento detalhado é ativado para as instâncias EC2 da frota de computação. Quando ativado, o console do Amazon EC2 exibe gráficos para monitorar as instâncias em intervalos de 1 minuto. Há custos adicionais quando esse recurso está ativado. O padrão é `false`.

Para obter mais informações, consulte [Habilitar ou desabilitar o monitoramento detalhado para instâncias](#) no Manual do usuário do Amazon EC2 para instâncias do Linux.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

Note

DetailedMonitoring é adicionado a partir da AWS ParallelCluster versão 3.6.0.

Alarms (Opcional)

CloudWatch Alarmes para o cluster.

[Política de atualização: essa configuração pode ser alterada durante uma atualização.](#)

Enabled (Opcional)

Set `true`, os CloudWatch alarmes para o cluster serão criados. O valor padrão é `true`.

Política de atualização: essa configuração pode ser alterada durante uma atualização.

Note

A partir da AWS ParallelCluster versão 3.8.0, os seguintes alarmes são criados para o Head Node: EC2 Health Check, utilização de CPU/memória/disco e um alarme composto incluindo todos os outros.

Seção **Tags**

(Opcional), Array define as tags que são usadas AWS CloudFormation e propagadas para todos os recursos do cluster. Para obter mais informações, consulte [Tag de recursos do AWS CloudFormation](#) no Guia do usuário do AWS CloudFormation .

Tags:

- Key: *string*
- Value: *string*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do **Tags**

Key (Obrigatório, String)

Define o nome da tag.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Value (Obrigatório, String)

Define o valor da tag.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Seção **AdditionalPackages**

(Opcional) Usado para identificar pacotes adicionais a serem instalados.

AdditionalPackages:

IntelSoftware:

IntelHpcPlatform: *boolean*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

IntelSoftware

(Opcional) Define a configuração das soluções selecionadas da Intel.

IntelSoftware:

IntelHpcPlatform: *boolean*

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Propriedades do IntelSoftware

IntelHpcPlatform (Opcional, Boolean)

Se `true`, indica que o [Contrato de licença de usuário final](#) do Intel Parallel Studio foi aceito. Isso faz com que o Intel Parallel Studio seja instalado no nó principal e compartilhado com os nós de computação. Isso adiciona vários minutos ao tempo que o nó principal leva para ser inicializado. A configuração `IntelHpcPlatform` só é compatível com o CentOS 7.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Seção DirectoryService

Note

Support for DirectoryService adicionado na AWS ParallelCluster versão 3.1.1.

(Opcional) As configurações do serviço de diretório para um cluster que oferece suporte ao acesso de vários usuários.

AWS ParallelCluster gerencia permissões que oferecem suporte ao acesso de vários usuários a clusters com um Active Directory (AD) por meio do Lightweight Directory Access Protocol (LDAP) suportado pelo [System Security Services Daemon](#) (SSSD). Para ter mais informações, consulte [O que é AWS Directory Service?](#) no Guia de administração do AWS Directory Service .

Recomendamos que você use LDAP sobre TLS/SSL (abreviado como LDAPS) para garantir que todas as informações potencialmente confidenciais sejam transmitidas por canais criptografados.

DirectoryService:

```
DomainName: string
DomainAddr: string
PasswordSecretArn: string
DomainReadOnlyUser: string
LdapTlsCaCert: string
LdapTlsReqCert: string
LdapAccessFilter: string
GenerateSshKeysForUsers: boolean
AdditionalSssdConfigs: dict
```

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

Propriedades do **DirectoryService**

Note

Se você planeja usar AWS ParallelCluster em uma única sub-rede sem acesso à Internet, consulte [AWS ParallelCluster em uma sub-rede única sem acesso à internet](#) os requisitos adicionais.

DomainName (Obrigatório, String)

O domínio do Active Directory (AD) que você usa para obter informações de identidade.

DomainName aceita os formatos de nome de domínio totalmente qualificado (FQDN) e nome distinto (DN) do LDAP.

- Exemplo de FQDN: corp.*example*.com
- Exemplo de LDAP DN: DC=*corp*, DC=*example*, DC=*com*

Essa propriedade corresponde ao parâmetro sssd-ldap que é chamado ldap_search_base.

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

DomainAddr (Obrigatório, String)

O URI ou URIs que apontam para o controlador de domínio AD usado como servidor LDAP. Essa URI corresponde ao parâmetro SSSD-LDAP que é chamado `ldap_uri`. O valor pode ser uma sequência de URIs separada por vírgula. Para usar o LDAP, você deve adicionar `ldap://` ao início de cada URI.

Valores de exemplo:

```
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                  # AWS ParallelCluster uses LDAPS by
  default
```

Se você usa LDAPS com verificação de certificado, os URIs devem ser nomes de host.

Se você usa LDAPS sem verificação de certificado ou LDAP, os URIs podem ser nomes de host ou endereços IP.

Use LDAP sobre TLS/SSL (LDAPS) para evitar a transmissão de senhas e outras informações confidenciais por canais não criptografados. Se AWS ParallelCluster não encontrar um protocolo, ele adicionará `ldaps://` ao início de cada URI ou nome de host.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

PasswordSecretArn (Obrigatório, String)

O Amazon Resource Name (ARN) do AWS Secrets Manager segredo que contém a senha em texto `DomainReadOnlyUser` simples. Esse conteúdo do segredo corresponde ao parâmetro SSSD-LDAP que é chamado `ldap_default_authtok`.

Note

Ao criar um segredo usando o AWS Secrets Manager console, certifique-se de selecionar “Outro tipo de segredo”, selecionar texto simples e incluir apenas o texto da senha no segredo.

Para obter mais informações sobre como usar AWS Secrets Manager para criar um segredo, consulte [Criar um AWS Secrets Manager segredo](#)

O cliente LDAP usa a senha para se autenticar no domínio AD como um `DomainReadOnlyUser` ao solicitar informações de identidade.

Se o usuário tiver a permissão para [DescribeSecret](#), `PasswordSecretArn` é validado. `PasswordSecretArn` é válido se o segredo especificado existir. Se a política do IAM do usuário não incluir `DescribeSecret`, `PasswordSecretArn` não for validada e uma mensagem de aviso será exibida. Para ter mais informações, consulte [Política básica de usuário pcluster do AWS ParallelCluster](#).

Quando o valor do segredo muda, o cluster não é atualizado automaticamente. Para atualizar o cluster para o novo valor secreto, você deve interromper a frota de computação com o comando [the section called "pcluster update-compute-fleet"](#) e, em seguida, executar o comando a seguir a partir do nó principal.

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

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

`DomainReadOnlyUser` (Obrigatório, String)

A identidade usada para consultar o domínio do AD para obter informações de identidade ao autenticar logins de usuários do cluster. Ele corresponde ao parâmetro SSSD-LDAP que é chamado `ldap_default_bind_dn`. Use suas informações de identidade do AD para esse valor.

Especifique a identidade no formato exigido pelo cliente LDAP específico que está no nó:

- MicrosoftAD:

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

- SimpleAD:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```


[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

LdapTlsCaCert (Opcional, String)

O caminho absoluto para um pacote de certificados contendo os certificados de cada autoridade de certificação na cadeia de certificação que emitiu um certificado para os controladores de domínio. Ele corresponde ao parâmetro SSSD-LDAP que é chamado `ldap_tls_cacert`.

Um pacote de certificados é um arquivo composto pela concatenação de certificados distintos no formato PEM, também conhecido como formato DER Base64 no Windows. Ele é usado para verificar a identidade do controlador de domínio AD que está atuando como servidor LDAP.

AWS ParallelCluster não é responsável pela colocação inicial dos certificados nos nós. Como administrador do cluster, você pode configurar o certificado no nó principal manualmente após a criação do cluster ou usar um [script de bootstrap](#). Como alternativa, é possível usar uma imagem de máquina da Amazon (AMI) que inclua o certificado configurado no nó principal.

O [Simple AD](#) não oferece suporte a LDAPS. Para saber como integrar um diretório Simple AD com AWS ParallelCluster, consulte [Como configurar um endpoint LDAPS para o Simple AD no blog](#) de AWS segurança.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

LdapTlsReqCert (Opcional, String)

Especifica quais verificações devem ser realizadas nos certificados do servidor em uma sessão TLS. Ele corresponde ao parâmetro SSSD-LDAP que é chamado `ldap_tls_reqcert`.

Valores válidos: `never`, `allow`, `try`, `demand` e `hard`.

`never`, `allow` e `try` permite que as conexões continuem mesmo se forem encontrados problemas com os certificados.

`demand` e `hard` permite que a comunicação continue se nenhum problema com os certificados for encontrado.

Se o administrador do cluster usar um valor que não exija que a validação do certificado seja bem-sucedida, uma mensagem de aviso será retornada ao administrador. Por motivos de segurança, recomendamos que você não desabilite a verificação do certificado.

O valor padrão é `hard`.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

LdapAccessFilter (Opcional, String)

Especifica um filtro para limitar o acesso ao diretório a um subconjunto de usuários. Essa propriedade corresponde ao parâmetro SSSD-LDAP que é chamado `ldap_access_filter`. Você pode usá-lo para limitar as consultas a um AD que ofereça suporte a um grande número de usuários.

Esse filtro pode bloquear o acesso do usuário ao cluster. No entanto, isso não afeta a capacidade de descoberta de usuários bloqueados.

Se essa propriedade for definida, o parâmetro SSSD `access_provider` será definido como `ldap` internamente pelo AWS ParallelCluster e não deverá ser modificado pelas configurações [DirectoryService](#) / [AdditionalSssdConfigs](#).

Se essa propriedade for omitida e o acesso personalizado do usuário não for especificado em [DirectoryService](#) / [AdditionalSssdConfigs](#), todos os usuários no diretório poderão acessar o cluster.

Exemplos:

```
!"(cn=SomeUser*)" # denies access to every user with alias starting with "SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias starting with "SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".
```

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

GenerateSshKeysForUsers (Opcional, Boolean)

Define se AWS ParallelCluster gera uma chave SSH para usuários do cluster imediatamente após a autenticação inicial no nó principal.

Se definido como `true`, uma chave SSH é gerada e salva `USER_HOME_DIRECTORY/.ssh/id_rsa`, se não existir, para cada usuário após a primeira autenticação no nó principal.

Para um usuário que ainda não foi autenticado no nó principal, a primeira autenticação pode ocorrer nos seguintes casos:

- O usuário faz login no nó principal pela primeira vez com sua própria senha.

- No nó principal, um sudoer muda para o usuário pela primeira vez: `su USERNAME`
- No nó principal, um sudoer executa um comando como usuário pela primeira vez: `su -u USERNAME COMMAND`

Os usuários podem usar a chave SSH para logins subsequentes no nó principal do cluster e nos nós de computação. Com AWS ParallelCluster, os logins com senha nos nós de computação do cluster são desativados por design. Se um usuário não estiver conectado ao nó principal, as chaves SSH não serão geradas e o usuário não poderá fazer login nos nós de computação.

O padrão é `true`.

[Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.](#)

`AdditionalSssdConfigs` (Opcional, Dict)

Um dicionário de pares de valores-chave contendo parâmetros e valores SSSD para gravar no arquivo de configuração SSSD em instâncias de cluster. Para obter uma descrição completa do arquivo de configuração SSSD, consulte as páginas do manual na instância SSSD e os arquivos de configuração relacionados.

Os parâmetros e valores do SSSD devem ser compatíveis com a configuração AWS ParallelCluster do SSSD, conforme descrito na lista a seguir.

- `id_provider` está definido como `ldap` internamente por AWS ParallelCluster e não deve ser modificado.
- `access_provider` é definido como `ldap` internamente AWS ParallelCluster quando [DirectoryService/LdapAccessFilter](#) é especificado, e essa configuração não deve ser modificada.

Se [DirectoryService](#) / [LdapAccessFilter](#) for omitido, sua especificação `access_provider` também será omitida. Por exemplo, se você definir `access_provider` como `simple` no [AdditionalSssdConfigs](#), então [DirectoryService](#) / [LdapAccessFilter](#) deverá ser especificado.

Os trechos de configuração a seguir são exemplos de configurações válidas para `AdditionalSssdConfigs`.

Este exemplo ativa o nível de depuração para registros SSSD, restringe a base de pesquisa a uma unidade organizacional específica e desativa o cache de credenciais.

```

DirectoryService:
  ...
  AdditionalSssdConfigs:
    debug_level: "0xFFF0"
    ldap_search_base: OU=Users,OU=CORP,DC=corp,DC=example,DC=com
    cache_credentials: False

```

Este exemplo especifica a configuração de um SSSD [simple](#) `access_provider`. Os usuários do `EngineeringTeam` têm acesso ao diretório. [DirectoryService](#) / [LdapAccessFilter](#) não deve ser definido neste caso.

```

DirectoryService:
  ...
  AdditionalSssdConfigs:
    access_provider: simple
    simple_allow_groups: EngineeringTeam

```

Política de atualização: a frota de computação deve ser interrompida para que essa configuração seja alterada para uma atualização.

Seção `DeploymentSettings`

Note

`DeploymentSettings` é adicionado a partir da AWS ParallelCluster versão 3.4.0.

(Opcional) Especifica a configuração das definições de implantação.

```

DeploymentSettings:
  LambdaFunctionsVpcConfig:
    SecurityGroupIds
      - string
    SubnetIds
      - string
  DisableSudoAccessForDefaultUser: Boolean
  DefaultUserHome: string # 'Shared' or 'Local'

```

Propriedades do **DeploymentSettings**

LambdaFunctionsVpcConfig

(Opcional) Especifica as configurações de VPC AWS Lambda das funções. Para ter mais informações, consulte [Configuração de VPC AWS Lambda no AWS ParallelCluster](#).

LambdaFunctionsVpcConfig:

SecurityGroupIds

- *string*

SubnetIds

- *string*

LambdaFunctionsVpcConfig properties

SecurityGroupIds (Obrigatório, [String])

A lista de IDs de grupos de segurança do Amazon VPC anexados às funções do Lambda.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

SubnetIds (Obrigatório, [String])

A lista de IDs de sub-rede anexadas às funções do Lambda.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

Os sub-redes e grupos de segurança devem estar na mesma VPC.

DisableSudoAccessForDefaultUser propriedade

Note

Essa opção de configuração só é compatível com Slurm Clusters.

(Opcional) Se `True`, os privilégios sudo do usuário padrão serão desativados. Isso se aplica a todos os nós no cluster.

```
# Main DeploymentSettings section in config yaml(applyes to HN, CF and LN)
DeploymentSettings:
  DisableSudoAccessForDefaultUser: True
```

Para atualizar o valor de `DisableSudoAccessForDefaultUser`, você deve interromper a frota de computação e todos os nós de login.

[Política de atualização: a frota computacional e os nós de login devem ser interrompidos para que essa configuração seja alterada para uma atualização.](#)

a propriedade `DefaultUserHome`

Quando definido como `Shared`, o cluster usará a configuração padrão e compartilhará o diretório do usuário padrão em todo o cluster por `/home/<default user>`.

Quando definidos como `Local`, o nó principal, os nós de login e os nós de computação terão, cada um, um diretório de usuário padrão local separado armazenado em `local/home/<default user>`.

Arquivos de configuração de imagem de compilação

AWS ParallelCluster a versão 3 usa arquivos YAML 1.1 para criar parâmetros de configuração de imagem. Confirme se o recuo está correto para reduzir os erros de configuração. Para obter mais informações, consulte a especificação do YAML 1.1 em <https://yaml.org/spec/1.1/>.

Esses arquivos de configuração são usados para definir como suas AWS ParallelCluster AMIs personalizadas são criadas usando o EC2 Image Builder. Os processos personalizados de criação de AMI são acionados usando o comando `pcluster build-image`. Para ver alguns exemplos de arquivos de configuração, consulte https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema.

Tópicos

- [Propriedades dos arquivos de configuração de imagem de compilação](#)
- [Seção Build](#)
- [Seção Image](#)
- [Seção DeploymentSettings](#)

Propriedades dos arquivos de configuração de imagem de compilação

Region (Opcional, String)

Especifica o Região da AWS para a build-image operação. Por exemplo, us-east-2.

Seção **Build**

(Obrigatório) Especifica a configuração na qual a imagem será criada.

```
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
```

Propriedades do **Build**

InstanceType (Obrigatório, String)

Especifica o tipo de instância usada para criar a imagem.

SubnetId (Opcional, String)

Especifica o ID de uma sub-rede existente na qual o servidor provisiona a instância para a criação da imagem. A sub-rede fornecida requer acesso à Internet.

Warning

O `pcluster build-image` usa a VPC padrão. Se a VPC padrão tiver sido excluída, talvez usando AWS Control Tower nossa AWS Landing Zone, a ID da sub-rede deverá ser especificada.

ParentImage (Obrigatório, String)

Especifica a imagem base. A imagem principal pode ser uma AWS ParallelCluster AMI não AMI ou uma AWS ParallelCluster AMI oficial para a mesma versão. Você não pode usar uma AMI AWS ParallelCluster oficial ou personalizada de uma versão diferente do AWS ParallelCluster. O formato deve ser o ARN de uma imagem `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion` ou uma ID de AMI `ami-12345678`.

SecurityGroupIds (Opcional, [String])

Especifica a lista de IDs de grupos de segurança da imagem.

Imds

Propriedades do Imds

(Opcional) Especifica as configurações do serviço de metadados de instância de ImageBuilder criação e teste (IMDS) do EC2.

Imds:

ImdsSupport: *string*

ImdsSupport (Opcional, String)

Especifica quais versões do IMDS são suportadas nas instâncias de ImageBuilder compilação e teste do EC2. Os valores compatíveis são `v2.0` e `v1.0`. O valor padrão é `v2.0`.

Se `ImdsSupport` estiver definido como `v1.0`, há compatibilidade com o IMDSv1 e o IMDSv2.

Se `ImdsSupport` estiver definido como `v2.0`, somente o IMDSv2 será suportado.

Para ter mais informações, consulte [Usar o IMDSv2](#), no Guia do usuário do EC2 para instâncias Linux.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

A partir da AWS ParallelCluster versão 3.7.0, o valor `ImdsSupport` padrão é `v2.0`. Recomendamos que você configure `ImdsSupport` para `v2.0` e substitua o IMDSv1 pelo IMDSv2 em suas chamadas de ações personalizadas.

O suporte para [Imds](#) foi [ImdsSupport](#) adicionado com a AWS ParallelCluster versão 3.3.0.

Iam

Propriedades do Iam

(Opcional) Especifica os recursos do IAM para a criação da imagem.

Iam:

```
InstanceRole: string
InstanceProfile: string
CleanupLambdaRole: string
AdditionalIamPolicies:
  - Policy: string
PermissionsBoundary: string
```

InstanceProfile (Opcional, String)

Especifica um perfil de instância para substituir o perfil de instância padrão para a instância do EC2 Image Builder. `InstanceProfile` e `InstanceRole` e `AdditionalIamPolicies` não podem ser especificados juntos. O formato é `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

InstanceRole (Opcional, String)

Especifica uma função de instância para substituir a função de instância padrão para a instância do EC2 Image Builder. `InstanceProfile` e `InstanceRole`

e `AdditionalIamPolicies` não podem ser especificados juntos. O formato é `arn:Partition:iam::Account:role/RoleName`.

`CleanupLambdaRole` (Opcional, String)

O ARN da função do IAM a ser usado para a AWS Lambda função de apoio ao recurso AWS CloudFormation personalizado que remove artefatos de construção na conclusão da compilação. O Lambda precisa ser configurado como entidade principal autorizada a assumir a função. O formato é `arn:Partition:iam::Account:role/RoleName`.

`AdditionalIamPolicies` (Opcional)

Especifica políticas adicionais do IAM a serem anexadas à instância do EC2 Image Builder usada para produzir a AMI personalizada.

```
AdditionalIamPolicies:  
- Policy: string
```

`Policy` (Opcional, [String])

Lista de políticas do IAM. O formato é `arn:Partition:iam::Account:policy/PolicyName`.

`PermissionsBoundary` (Opcional, String)

O ARN da política do IAM usado como limite de permissões para todos os perfis do IAM criados pelo AWS ParallelCluster. Para obter mais informações sobre limites de permissões do IAM, consulte [Limites de permissões para entidades do IAM](#) no Guia do usuário do IAM. O formato é `arn:Partition:iam::Account:policy/PolicyName`.

Components

Propriedades do **Components**

(Opcional) Especifica ImageBuilder os componentes do EC2 a serem usados durante o processo de criação da AMI, além dos fornecidos por padrão pela AWS ParallelCluster. Esses componentes podem ser usados para personalizar o processo de criação da AMI. Para ter mais informações, consulte [AWS ParallelCluster Personalização da AMI](#).

```
Components:  
- Type: string  
  Value: string
```

Type (Opcional, String)

Especifica o tipo do par tipo-valor para o componente. O tipo pode ser `arn` ou `script`.

Value (Opcional, String)

Especifica o valor do par tipo-valor para o componente. Quando o tipo é `arn`, esse é o ARN de um componente do EC2 Image Builder. Quando o tipo é `script`, esse é o link `https` ou `s3` que aponta para o script a ser usado ao criar o componente EC2 Image Builder.

Tags

Propriedades do **Tags**

(Opcional) Especifica a lista de tags a serem definidas nos recursos usados para criar a AMI.

Tags:

- Key: *string*
- Value: *string*

Key (Opcional, String)

Define o nome da tag.

Value (Opcional, String)

Define o valor da tag.

UpdateOsPackages

Propriedades do **UpdateOsPackages**

(Opcional) Especifica se o sistema operacional é atualizado antes da instalação da pilha AWS ParallelCluster de software.

UpdateOsPackages:

- Enabled: *boolean*

Enabled (Opcional, Boolean)

Set `true`, o sistema operacional for atualizado e reinicializado antes de instalar o AWS ParallelCluster software. O padrão é `false`.

Note

Quando `UpdateOsPackages` é ativado, todos os pacotes de sistema operacional disponíveis são atualizados, incluindo o kernel. Como cliente, você é responsável por verificar se a atualização é compatível com as dependências da AMI que não estão incluídas na atualização.

Por exemplo, suponha que você esteja criando uma AMI para a AWS ParallelCluster versão X.0 que vem com a versão Y.0 do kernel e a versão Z.0 de algum componente. Suponha que a atualização disponível inclua a versão atualizada do kernel Y.1 sem atualizações para o componente Z.0. Antes de habilitar `UpdateOsPackages`, é sua responsabilidade verificar se o componente Z.0 é compatível com o kernel Y.1.

Seção **Image**

(Opcional) Define as propriedades da imagem para a criação da imagem.

Image:

Name: *string*

RootVolume:

Size: *integer*

Encrypted: *boolean*

KmsKeyId: *string*

Tags:

- **Key:** *string*

Value: *string*

Propriedades do **Image**

Name (Opcional, String)

Especifica o nome da AMI. Se não for especificado, o nome usado ao chamar o comando [pcluster build-image](#) será usado.

Tags

Propriedades do **Tags**

(Opcional) Especifica pares de chave-valor para a imagem.

Tags:

- **Key:** *string*
Value: *string*

Key (Opcional, String)

Define o nome da tag.

Value (Opcional, String)

Define o valor da tag.

RootVolume

Propriedades do **RootVolume**

(Opcional) Especifica as propriedades do volume raiz da imagem.

RootVolume:

- Size:** *integer*
- Encrypted:** *boolean*
- KmsKeyId:** *string*

Size (Opcional, Integer)

Especifica o tamanho do volume raiz da imagem em GiB. O tamanho padrão é o tamanho de [ParentImage](#) mais 27 GiB.

Encrypted (Opcional, Boolean)

Especifica se o volume está criptografado. O valor padrão é `false`.

KmsKeyId (Opcional, String)

Especifica o ARN da chave usada para criptografar AWS KMS o volume. O formato é `"arn:Partition:kms:Region:Account:key/KeyId"`.

Seção **DeploymentSettings**

(Opcional) Especifica a configuração das definições de implantação.

DeploymentSettings:

```
LambdaFunctionsVpcConfig:  
  SecurityGroupIds  
    - string  
  SubnetIds  
    - string
```

Propriedades do **DeploymentSettings**

LambdaFunctionsVpcConfig

(Opcional) Especifica as configurações de VPC AWS Lambda das funções. Para ter mais informações, consulte [Configuração de VPC AWS Lambda no AWS ParallelCluster](#).

```
LambdaFunctionsVpcConfig:  
  SecurityGroupIds  
    - string  
  SubnetIds  
    - string
```

LambdaFunctionsVpcConfig properties

SecurityGroupIds (Obrigatório, [String])

A lista de IDs de grupos de segurança do Amazon VPC anexados às funções do Lambda.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

SubnetIds (Obrigatório, [String])

A lista de IDs de sub-rede anexadas às funções do Lambda.

Política de atualização: se essa configuração for alterada, a atualização não será permitida.

Note

Os sub-redes e grupos de segurança devem estar na mesma VPC.

Note

DeploymentSettings é adicionado a partir da AWS ParallelCluster versão 3.4.0.

Referência da API do AWS ParallelCluster

Esta seção fornece descrições, sintaxe e exemplos de uso para cada uma das ações da API do AWS ParallelCluster.

Tópicos

- [buildImage](#)
- [createCluster](#)
- [deleteCluster](#)
- [deleteClusterInstances](#)
- [deleteImage](#)
- [describeCluster](#)
- [describeClusterInstances](#)
- [describeComputeFleet](#)
- [describeImage](#)
- [getClusterLogEvents](#)
- [getClusterStackEvents](#)
- [getImageLogEvents](#)
- [getImageStackEvents](#)
- [listClusters](#)
- [listClusterLogStreams](#)
- [listImageLogStreams](#)
- [listImages](#)
- [listOfficialImages](#)
- [updateCluster](#)
- [updateComputeFleet](#)

buildImage

Cria uma imagem personalizada do AWS ParallelCluster em uma Região da AWS.

Tópicos

- [Sintaxe da solicitação](#)

- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
POST /v3/images/custom
{
  "imageConfiguration": "string",
  "imageId": "string",
  "dryrun": boolean,
  "region": "string",
  "rollbackOnFailure": boolean,
  "supressValidators": [ "string" ],
  "validationFailureLevel": "string"
}
```

Corpo da solicitação

imageConfiguration

A configuração da imagem como um documento YAML.

Tipo: string

Obrigatório: sim

imageId

A ID da imagem a ser parada.

Tipo: string

Obrigatório: sim

dryrun

Se definido como `true`, executa somente a validação da solicitação, sem criar nenhum recurso. Use este parâmetro para validar a configuração da imagem. O padrão é `false`.

Tipo: booliano

Obrigatório: não

region

A Região da AWS em que você executa o comando para criar a imagem.

Tipo: string

Obrigatório: não

rollbackOnFailure

Se definido como `true`, ocorre reversão da pilha de imagens em caso de falha na criação da imagem. O padrão é `false`.

Tipo: booleano

Obrigatório: não

suppressValidators

Identifica um ou mais validadores de configuração a serem suprimidos.

Tipo: lista de strings

Formato: (ALL | type: [A-Za-z0-9]+)

Obrigatório: não

validationFailureLevel

O nível mínimo de validação que faz com que a criação da imagem falhe. O padrão é `ERROR`.

Tipo: string

Valores válidos: `INFO` | `WARNING` | `ERROR`

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

image

imageId

O ID da imagem.

Tipo: string

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

ec2AmiInfo

ami-id

O ID da AMI do EC2.

Tipo: string

imageBuildStatus

O status de construção da imagem.

Tipo: string

Valores válidos: BUILD_IN_PROGRESS | BUILD_FAILED | BUILD_COMPLETE |
DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE

region

A Região da AWS que a imagem está embutida.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar a imagem.

Tipo: string

validationMessages

Uma lista de mensagens com um nível de validação inferior a `validationFailureLevel`. A lista de mensagens é coletada durante a validação da configuração.

id

O ID do validador.

Tipo: string

level

O nível de validação.

Tipo: string

Valores válidos: INFO | WARNING | ERROR

message

Mensagem de validação.

Tipo: string

type

O tipo de validador.

Tipo: string

Exemplo

Python

Solicitação

```
$ build_image(custom-image-id, custom-image-config.yaml)
```

Resposta 200

```
{
  '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

Cria um cluster gerenciado em uma Região da AWS.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
POST /v3/clusters
{
  "clusterName": "string",
  "clusterConfiguration": "string",
  "dryrun": boolean,
  "region": "string",
  "rollbackOnFailure": boolean,
  "suppressValidators": [ "string" ],
  "validationFailureLevel": "string"
}
```

Corpo da solicitação

clusterConfiguration

A configuração do cluster como um documento YAML.

Tipo: string

Obrigatório: sim

clusterName

O nome do cluster a ser criado.

O nome deve iniciar com um caractere alfabético. O nome do pode ter até 60 caracteres. Se a contabilidade do Slurm estiver ativada, o nome poderá ter até 40 caracteres.

Tipo: string

Obrigatório: sim

dryrun

Se definido como `true`, executa somente a validação da solicitação, sem criar nenhum recurso. Use este parâmetro para validar a configuração do cluster. O padrão é `false`.

Tipo: booleano

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

rollbackOnFailure

Se definido como `true`, ocorre reversão da pilha de cluster em caso de falha na criação do cluster. O padrão é `true`.

Tipo: booleano

Obrigatório: não

suppressValidators

Identifica um ou mais validadores de configuração a serem suprimidos.

Tipo: lista de strings

Formato: (ALL | type: [A-Za-z0-9]+)

Obrigatório: não

validationFailureLevel

O nível mínimo de validação que faz com que a criação de cluster falhe. O padrão é `ERROR`.

Tipo: string

Valores válidos: INFO | WARNING | ERROR

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

clusterName

Nome do cluster.

Tipo: string

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Tipo: string

Valores válidos: 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

clusterStatus

Tipo: string

Valores válidos: CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE
| DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE |
UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED

region

A Região da AWS em que o cluster foi criado.

Tipo: string

programador

metadados

Os metadados do programador

name

O nome do programador.

Tipo: string

version

A versão do programador.

Tipo: string

type

O tipo de programador.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar o cluster.

Tipo: string

validation_messages

Uma lista de mensagens com um nível de validação inferior a `validationFailureLevel`. A lista de mensagens é coletada durante a validação da configuração.

id

O ID do validador.

Tipo: string

level

Tipo: string

Valores válidos: INFO | WARNING | ERROR

message

Mensagem de validação.

Tipo: string

type

O tipo do validador.

Tipo: string

Exemplo

Python

Solicitação

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

Resposta 200

```
{
  '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'
}
}
```

deleteCluster

Inicia a exclusão de um cluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
DELETE /v3/clusters/{clusterName}
{
  "region": "string"
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

region

A Região da AWS em que o cluster é excluído.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

cluster

Uma lista de instâncias de cluster

clusterName

Nome do cluster.

Tipo: string

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Tipo: string

Valores válidos: 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

clusterStatus

Tipo: string

Valores válidos: CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE
| DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE |
UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED

region

A Região da AWS em que o cluster foi criado.

Tipo: string

programador

metadados

Os metadados do programador.

name

O nome do programador.

Tipo: string

version

A versão do programador.

Tipo: string

type

O tipo de programador.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar o cluster.

Tipo: string

Exemplo

Python

Solicitação

```
$ delete_cluster(cluster_name_3x)
```

Resposta 200

```
{
  'cluster': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-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

Inicia o encerramento forçado de todos os nós de computação do cluster. Essa ação não é compatível com clusters AWS Batch.

Tópicos

- [Sintaxe da solicitação](#)

- [Corpo da solicitação](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
DELETE /v3/clusters/{clusterName}/instances
{
  "force": boolean,
  "region": "string"
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

force

Se definido como `true`, força a exclusão quando o cluster com o nome fornecido não for encontrado. O padrão é `false`.

Tipo: booliano

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Corpo da resposta

Nenhum

Exemplo

Python

Solicitação

```
$ delete_cluster_instances(cluster_name_3x)
```

Resposta 200

Nenhum

deletelImage

Inicia a exclusão da imagem personalizada do AWS ParallelCluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
DELETE /v3/images/custom/{imageId}  
{  
  "force": boolean,  
  "region": "string"  
}
```

Corpo da solicitação

imageId

O ID da imagem.

Tipo: string

Obrigatório: sim

force

Se definido como `true`, força a exclusão da AMI. Use esse parâmetro se houver instâncias usando a AMI ou se a AMI for compartilhada. O padrão é `false`.

Tipo: booleano

Obrigatório: não

region

A Região da AWS em que a imagem foi criada.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "image": {
    "imageId": "string",
    "ec2AmiInfo": {
      "amiId": "string"
    },
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "imageBuildStatus": "DELETE_IN_PROGRESS",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS"
  }
}
```

Corpo da resposta

image

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

ec2AmiInfo

amild

O ID da AMI do EC2.

Tipo: string

imageBuildStatus

O status de construção da imagem.

Tipo: string

Valores válidos: BUILD_IN_PROGRESS | BUILD_FAILED | BUILD_COMPLETE |
DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE

imageId

O ID da imagem.

Tipo: string

region

A Região da AWS em que a imagem é criada.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar a imagem.

Tipo: string

Exemplo

Python

Solicitação

```
$ delete_image(custom-image-id)
```

Resposta 200

```
{
  'image': {
    'image_build_status': 'DELETE_IN_PROGRESS',
    'image_id': 'custom-image-id',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

describeCluster

Obtém informações detalhadas sobre um cluster existente.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters/{clusterName}
{
  "region": "string"
```

```
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

Note

`failureReason` mudou para `failures` a partir do AWS ParallelCluster versão 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",
  "scheme": "string",
  "healthyNodes": integer,
  "unhealthyNodes": integer
}
}
```

Corpo da resposta

clusterName

O nome do cluster.

Tipo: string

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

clusterConfiguration

url

URL do arquivo de configuração do cluster.

Tipo: string

clusterStatus

Status do cluster.

Tipo: string

Valores válidos: CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE
| DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE |
UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED

computeFleetStatus

Status da frota de computação.

Tipo: string

Valores válidos: START_REQUESTED | STARTING | RUNNING | PROTECTED |
STOP_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

creationTime

Marca de data e hora da criação do cluster.

Tipo: datetime

lastUpdatedTime

Marca de data/hora da última atualização do cluster.

Tipo: datetime

region

A Região da AWS em que o cluster foi criado.

Tipo: string

tags

Lista de tags associadas ao cluster.

chave

Nome da tag.

Tipo: string

tag

Valor da tag.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar o cluster.

Tipo: string

falhas

A lista de falhas quando a pilha do cluster está em status CREATE_FAILED.

failureCode

Código de falha quando a pilha do cluster está em status CREATE_FAILED.

Tipo: string

failureReason

Razão da falha quando a pilha do cluster está em status CREATE_FAILED.

Tipo: string

head_node

Nó principal do cluster.

instanceId

O ID da instância do EC2.

Tipo: string

instanceType

Um tipo de instância do EC2.

Tipo: string

launchTime

A hora em que a instância do EC2 foi iniciada.

Tipo: datetime

privateIpAddress

O endereço IP privado do cluster.

Tipo: string

publicIpAddress

O endereço IP público do cluster.

Tipo: string

estado

O status da instância do nó principal.

Tipo: string

Valores válidos: `pending` | `running` | `shutting-down` | `terminated` | `stopping`
| `stopped`

programador

metadados

Os metadados do programador.

name

O nome do programador.

Tipo: string

version

A versão do programador.

Tipo: string

loginNodes**status**

O status do nó de login.

Tipo: string

Valores válidos: PENDING | FAILED | ACTIVE

address

O endereço do nó de login.

Tipo: string

scheme

O esquema do nó de login.

Tipo: string

scheme

Número de nós saudáveis.

Tipo: inteiro

scheme

O número de nós não saudáveis.

Tipo: inteiro

type

O tipo de programador.

Tipo: string

Exemplo

Python

Solicitação

```
$ describe_cluster(cluster_name_3x)
```

Resposta 200

```
{
  '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

Descreve as instâncias que pertencem a um cluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters/{clusterName}/instances  
{  
  "nextToken": "string",  
  "nodeType": "string",  
  "queueName": "string",  
  "region": "string"  
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

nodeType

Filtra as instâncias por tipo de nó.

Tipo: string

Valores válidos: HeadNode, ComputeNode, LoginNode

Obrigatório: não

queueName

Filtra as instâncias pelo nome da fila.

Tipo: string

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "nextToken": "string",
  "instances": [
    {
      "instanceId": "string",
      "instanceType": "string",
      "launchTime": "2019-08-24T14:15:22Z",
      "privateIpAddress": "string",
      "publicIpAddress": "string",
      "state": "pending",
      "nodeType": "HeadNode",
      "queueName": "string"
    }
  ]
}
```

```
}  
]  
}
```

Corpo da resposta

instâncias

A lista de instâncias do cluster.

instanceld

O ID da instância do EC2.

Tipo: string

instanceType

O tipo de instância do EC2.

Tipo: string

launchTime

A hora em que a instância do EC2 foi iniciada.

Tipo: datetime

nodeType

O tipo de nó.

Tipo: string

Valores válidos: HeadNode, ComputeNode, LoginNode

publicIpAddress

O endereço IP público do cluster.

Tipo: string

queueName

O nome da fila na qual a instância do EC2 está apoiando um nó.

Tipo: string

estado

O status da instância EC2 do nó.

Tipo: string

Valores válidos: `pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ describe_cluster_instances(cluster_name_3x)
```

Resposta 200

```
{
  '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

Descreve o status da frota de computação.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters/{clusterName}/computefleet
{
  "region": "string"
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "status": "START_REQUESTED",
  "lastStatusUpdatedTime": "2019-08-24T14:15:22Z"
}
```

Corpo da resposta

status

Tipo: string

Valores válidos: START_REQUESTED | STARTING | RUNNING | PROTECTED | STOP_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

lastStatusUpdatedTime

A marca de data e hora que representa a hora da última atualização de status.

Tipo: datetime

Exemplo

Python

Solicitação

```
$ describe_compute_fleet(cluster_name_3x)
```

Resposta 200

```
{
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
  tzinfo=tzlocal()),
  'status': 'RUNNING'
}
```

describelmage

Obtém informações detalhadas sobre uma imagem existente.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)

- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/images/custom/{imageId}
{
  "region": "string"
}
```

Corpo da solicitação

imageId

O ID da imagem.

Tipo: string

Obrigatório: sim

region

A Região da AWS em que a imagem foi criada.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

imageId

O ID da imagem para a qual recuperar informações detalhadas.

Tipo: string

imageBuildStatus

O status de construção da imagem.

Tipo: string

Valores válidos: BUILD_IN_PROGRESS | BUILD_FAILED | BUILD_COMPLETE | DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE

imageConfiguration

url

URL do arquivo de configuração da imagem.

Tipo: string

region

A Região da AWS em que a imagem é criada.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar a imagem.

Tipo: string

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackCreationTime

A marca de data e hora de quando a pilha do CloudFormation foi criada.

Tipo: datetime

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

cloudformationStackStatusReason

A razão do status da pilha do CloudFormation.

Tipo: string

cloudformationStackTags

A lista de tags para a pilha do CloudFormation.

chave

O nome da tag.

Tipo: string

value

O valor da tag.

Tipo: string

creationTime

A marca de data/hora da criação da imagem.

Tipo: datetime

ec2AmiInfo

amild

O ID da AMI do EC2.

Tipo: string

amiName

O nome da AMI do EC2.

Tipo: string

do seu aplicativo

A arquitetura da AMI do EC2.

Tipo: string

estado

O estado da AMI do EC2.

Tipo: string

Valores válidos: PENDING | AVAILABLE | INVALID | DEREGISTERED | TRANSIENT
| FAILED | ERROR

tags

Lista de tags da AMI do EC2.

chave

Nome da tag.

Tipo: string

value

Valor da tag.

Tipo: string

imagebuilderImageStatus

O status do ImageBuilder.

Tipo: string

Valores válidos: PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING |
INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

imagebuilderImageStatusReason

A razão do status da imagem do ImageBuilder.

Tipo: string

imageBuildLogsArn

O nome do recurso da Amazon (ARN) dos logs do processo de criação de imagens.

Tipo: string

Exemplo

Python

Solicitação

```
$ describe_image(custom-image-id)
```

Resposta 200

```
{
  '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:s3_bucket',
      'value': 'parallelcluster-abcdef01234567890-v1-do-not-delete'
    },
    {
      '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://parallelcluster-abcdef01234567890-v1-do-not-delete/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://parallelcluster-abcdef01234567890-v1-do-not-delete.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'
}
```

getClusterLogEvents

Recupera os eventos que estão associados a um fluxo de logs.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters/{clusterName}/logstreams/{logStreamName}
{
  "endTime": datetime,
  "limit": float,
  "nextToken": "string",
  "region": "string",
  "startFromHead": boolean,
```

```
"startTime": datetime
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

logStreamName

O nome do fluxo de logs.

Tipo: string

Obrigatório: sim

endTime

O final do intervalo de tempo, expresso no formato ISO 8601. Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos.

Tipo: datetime

Formato: 2021-01-01T20:00:00Z

Obrigatório: não

limit

O número máximo de eventos de log retornados. Se você não especificar um valor, o máximo será o máximo de eventos de log que couberem em um tamanho de resposta de 1 MB, com até 10.000 eventos de log.

Tipo: flutuante

Obrigatório: não

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

startFromHead

Se definido como `true`, os eventos de log mais antigos serão retornados primeiro. Se o valor for `false`, os eventos de log mais recentes serão retornados primeiro. O padrão é `false`.

Tipo: booleano

Obrigatório: não

startTime

O início do intervalo de tempo, expresso no formato ISO 8601. Eventos com uma marca de data/hora igual ou posterior a esse horário estão incluídos.

Tipo: datetime

Formato: `2021-01-01T20:00:00Z`

Obrigatório: não

Sintaxe da resposta

```
{
  "nextToken": "string",
  "prevToken": "string",
  "events": [
    {
      "timestamp": "2019-08-24T14:15:22Z",
      "message": "string"
    }
  ]
}
```


Corpo da resposta

eventos

Lista de eventos filtrados.

message

A mensagem do evento.

Tipo: string

timestamp

O timestamp do evento.

Tipo: datetime

nextToken

Um token usado para solicitações paginadas.

Tipo: string

prevToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ get_cluster_log_events(cluster_name_3x, log_stream_name=ip-192-0-2-26.i-  
abcdef01234567890.cfn-init)
```

Resposta 200

```
"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"
  },
  ...
]
```

getClusterStackEvents

Recupera os eventos que estão associados a uma pilha de um cluster.

Note

A partir da versão 3.6.0, o AWS ParallelCluster usa pilhas aninhadas para criar os recursos associados às filas e aos recursos de computação. A API `GetClusterStackEvents` e o comando `pcluster get-cluster-stack-events` retornam apenas os eventos da pilha principal do cluster. Você pode visualizar os eventos da pilha de clusters, incluindo aqueles relacionados a filas e recursos de computação, no console do CloudFormation.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters/{clusterName}/stackevents
{
  "nextToken": "string",
  "region": "string"
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

eventos

Lista de eventos filtrados.

clientRequestToken

O token passado para a ação que gerou esse evento.

Tipo: string

eventId

O ID exclusivo desse evento.

Tipo: string

logicalResourceId

O nome lógico do recurso, conforme definido no modelo.

Tipo: string

physicalResourceId

O nome ou o identificador exclusivo que está associado à instância física do recurso.

Tipo: string

resourceProperties

Um BLOB das propriedades usadas para criar o recurso.

Tipo: string

resourceStatus

O status do recurso.

Tipo: string

Valores válidos: 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

Uma mensagem de sucesso ou falha associada ao recurso.

Tipo: string

resourceType

O tipo de recurso.

Tipo: string

stackId

O nome de ID exclusivo da instância da pilha.

Tipo: string

stackName

O nome que está associado a uma pilha.

Tipo: string

timestamp

A hora em que o status foi atualizado.

Tipo: datetime

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ get_cluster_stack_events(cluster_name_3x)
```

Resposta 200

```
{
  '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())
    },
    ...
  ]
}
```

getImageLogEvents

Recupera os eventos que estão associados a uma criação de imagem.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/images/custom/{imageId}/logstreams/{logStreamName}
{
```

```
"endTime": datetime,  
"limit": float,  
"nextToken": "string",  
"region": "string",  
"startFromHead": boolean,  
"startTime": datetime  
}
```

Corpo da solicitação

imageId

O ID da imagem.

Tipo: string

Obrigatório: sim

logStreamName

O nome do fluxo de logs.

Tipo: string

Obrigatório: sim

endTime

O final do intervalo de tempo, expresso no formato ISO 8601. Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos.

Tipo: datetime

Formato: 2021-01-01T20:00:00Z

Obrigatório: não

limit

O número máximo de eventos de log retornados. Se você não especificar um valor, o máximo será o máximo de eventos de log que couberem em um tamanho de resposta de 1 MB, com até 10.000 eventos de log.

Tipo: flutuante

Obrigatório: não

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que a imagem está.

Tipo: string

Obrigatório: não

startFromHead

Se definido como `true`, retorna os eventos de log mais antigos primeiro. Se definido como `false`, retorna os eventos de log mais recentes primeiro. O padrão é `false`.

Tipo: booleano

Obrigatório: não

startTime

O início do intervalo de tempo, expresso no formato ISO 8601. Eventos com uma marca de data/hora igual ou posterior a esse horário estão incluídos.

Tipo: datetime

Formato: `2021-01-01T20:00:00Z`

Obrigatório: não

Sintaxe da resposta

```
{
  "nextToken": "string",
  "prevToken": "string",
  "events": [
    {
```



```
    "timestamp": "2019-08-24T14:15:22Z",  
    "message": "string"  
  }  
]  
}
```

Corpo da resposta

eventos

Lista de eventos filtrados.

message

A mensagem do evento.

Tipo: string

timestamp

O timestamp do evento.

Tipo: datetime

nextToken

Um token usado para solicitações paginadas.

Tipo: string

prevToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

Resposta 200

```
"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"  
  },  
  ...  
]
```

getImageStackEvents

Recupera os eventos estão associados à pilha de uma criação de imagem.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/images/custom/{imageId}/stackevents  
{  
  "nextToken": "string",  
  "region": "string"  
}
```

Corpo da solicitação

imageId

O ID da imagem.

Tipo: string

Obrigatório: sim

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que a imagem está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

events

Lista de eventos filtrados.

clientRequestToken

O token passado para a ação que gerou esse evento.

Tipo: string

eventId

O ID exclusivo desse evento.

Tipo: string

logicalResourceId

O nome lógico do recurso, conforme definido no modelo.

Tipo: string

physicalResourceId

O nome ou o identificador exclusivo que está associado à instância física do recurso.

Tipo: string

resourceProperties

Um BLOB das propriedades usadas para criar o recurso.

Tipo: string

resourceStatus

O status do recurso.

Tipo: string

Valores válidos: 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

Uma mensagem de sucesso ou falha associada ao recurso.

Tipo: string

resourceType

O tipo de recurso.

Tipo: string

stackId

O nome de ID exclusivo da instância da pilha.

Tipo: string

stackName

O nome que está associado a uma pilha.

Tipo: string

timestamp

A hora em que o status foi atualizado.

Tipo: datetime

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ get_image_stack_events(image_id)
```

Resposta 200

```
{  
  '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())
    },
    ...
]
}

```

listClusters

Retorna uma lista de clusters existentes.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)

- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters
{
  "clusterStatus": "string",
  "nextToken": "string",
  "region": "string"
}
```

Corpo da solicitação

clusterStatus

Filtro por status do cluster. O padrão é todos os clusters.

Tipo: string

Valores válidos: CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE
| DELETE_IN_PROGRESS | DELETE_FAILED | UPDATE_IN_PROGRESS |
UPDATE_COMPLETE | UPDATE_FAILED

Obrigatório: não

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS dos clusters.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

clusters

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

clusterName

O nome do cluster.

Tipo: string

clusterStatus

Status do cluster.

Tipo: string

Valores válidos: CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE
| DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE |
UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED

programador

metadados

Os metadados do programador.

name

O nome do programador.

Tipo: string

version

A versão do programador.

Tipo: string

type

O tipo de programador.

Tipo: string

region

A Região da AWS em que o cluster foi criado.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar o cluster.

Tipo: string

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ list_clusters()
```

Resposta 200

```
{
  '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'
    },
    ...
  ]
}
```

listClusterLogStreams

Recupera a lista de fluxo de logs que estão associados a uma um cluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/clusters/{clusterName}/logstreams
{
  "filters": [ "string" ],
  "nextToken": "string",
  "region": "string"
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

filtros

Filtra os fluxos de log.

Os filtros aceitos são:

- `private-dns-name`: a forma abreviada do nome DNS privado da instância (por exemplo, `ip-10-0-0-101`).
- `node-type`: Valor válido: `HeadNode`.

Tipo: Matriz de strings exclusiva

Formato: `Name=a, Values=1 Name=b, Values=2, 3`

Obrigatório: não

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

logStreams

Uma lista de fluxo de logs

creationTime

A hora em que o fluxo foi criado.

Tipo: datetime

firstEventTimestamp

A hora do primeiro evento do fluxo.

Tipo: datetime

lastEventTimestamp

A hora do último evento do fluxo. O valor de LastEventTime é atualizado com base em uma consistência eventual. Normalmente, ele é atualizado em menos de uma hora após a ingestão, mas em situações raras, pode demorar mais.

Tipo: datetime

lastIngestionTime

O último horário de ingestão.

Tipo: datetime

logStreamArn

O nome do recurso da Amazon (ARN) do fluxo de logs.

Tipo: string

logStreamName

O nome do fluxo de logs.

Tipo: string

uploadSequenceToken

O token de sequência.

Tipo: string

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ list_cluster_log_streams(cluster_name_3x)
```

Resposta 200

```
{
  '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': '####'
    },
    ...
  ]
}
```

listImageLogStreams

Recupera a lista de fluxos de log associados a uma imagem.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)

- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/images/custom/{imageId}/logstreams
{
  "nextToken": "string",
  "region": "string"
}
```

Corpo da solicitação

imageId

O ID da imagem.

Tipo: string

Obrigatório: sim

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que a imagem está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

logStreams

Uma lista de fluxo de logs.

creationTime

A hora em que o fluxo foi criado.

Tipo: datetime

firstEventTimestamp

A hora do primeiro evento do fluxo.

Tipo: datetime

lastEventTimestamp

A hora do último evento do fluxo. O valor de LastEventTime é atualizado com base em uma consistência eventual. Normalmente, ele é atualizado em menos de uma hora após a ingestão, mas em situações raras, pode demorar mais.

Tipo: datetime

lastIngestionTime

O último horário de ingestão.

Tipo: datetime

logStreamArn

O nome do recurso da Amazon (ARN) do fluxo de logs.

Tipo: string

logStreamName

O nome do fluxo de logs.

Tipo: string

uploadSequenceToken

O token de sequência.

Tipo: string

next_token

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ list_image_log_streams(custom-image-id)
```

Resposta 200

```
{
  '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': '####'
  },
  ...
]
}
```

listImages

Recupera a lista de imagens personalizadas existentes.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /images/custom
{
  "imageStatus": "string",
  "nextToken": "string",
  "region": "string"
}
```

Corpo da solicitação

imageStatus

Filtra as imagens pelo status fornecido.

Tipo: string

Valores válidos: AVAILABLE | PENDING | FAILED

Obrigatório: Sim

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Obrigatório: não

region

A Região da AWS em que as imagens estão.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "nextToken": "string",
  "images": [
    {
      "imageId": "string",
      "ec2AmiInfo": {
        "amiId": "string"
      },
      "region": "string",
      "version": "string",
      "cloudformationStackArn": "string",
      "imageBuildStatus": "BUILD_IN_PROGRESS",
      "cloudformationStackStatus": "CREATE_IN_PROGRESS"
    }
  ]
}
```

Corpo da resposta

images

Uma lista de imagens.

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

ec2AmiInfo

ami-id

O ID da AMI do EC2.

Tipo: string

imageBuildStatus

O status de construção da imagem.

Valores válidos: BUILD_IN_PROGRESS | BUILD_FAILED | BUILD_COMPLETE |
DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE

Tipo: string

imageId

O ID da imagem.

Tipo: string

region

A Região da AWS em que a imagem é criada.

Tipo: string

version

A versão do AWS ParallelCluster usada para criar a imagem.

Tipo: string

nextToken

Um token usado para solicitações paginadas.

Tipo: string

Exemplo

Python

Solicitação

```
$ list_images("AVAILABLE")
```

Resposta 200

```
{
  '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 a lista de imagens oficiais do AWS ParallelCluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
GET /v3/images/official
{
  "architecture": "string",
  "os": "string",
  "region": "string"
}
```

Corpo da solicitação

do seu aplicativo

Filtra por arquitetura. O padrão é sem filtro.

Tipo: string

Valores válidos: x86_64 | arm64

Obrigatório: não

os

Filtra por distribuição do sistema operacional. O padrão é sem filtro.

Tipo: string

Valores válidos: alinux2 | centos7 | ubuntu2204 | ubuntu2004 | rhe18

Obrigatório: não

region

A Região da AWS em que as imagens oficiais estão listadas.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "images": [
    {
      "architecture": "string",
      "amiId": "string",
      "name": "string",
      "os": "string",
      "version": "string"
    }
  ]
}
```

Corpo da resposta

images

amiId

O ID da AMI.

Tipo: string

do seu aplicativo

A arquitetura da AMI.

Tipo: string

name

O nome da AMI.

Tipo: string

os

O sistema operacional da AMI.

Tipo: string

version

A versão do AWS ParallelCluster.

Tipo: string

Exemplo

Python

Solicitação

```
$ list_official_images()
```

Resposta 200

```
{
  'images': [
    {
      'ami_id': 'ami-015cfef4e0d6306b2',
      'architecture': 'x86_64',
      'name': 'aws-parallelcluster-3.2.1-ubuntu-2004-lts-hvm-x86_64-202202261505 '
      '2022-02-26T15-08-34.759Z',
      'os': 'ubuntu2004',
      'version': '3.2.1'
    },
    ...
  ]
}
```

updateCluster

Atualiza um cluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)

- [Exemplo](#)

Sintaxe da solicitação

```
PUT /v3/clusters/{clusterName}
{
  "clusterConfiguration": "string",
  "dryrun": boolean,
  "forceUpdate": boolean,
  "region": "string",
  "suppressValidators": "string",
  "validationFailureLevel": "string"
}
```

Corpo da solicitação

clusterConfiguration

A configuração do cluster como um documento YAML.

Obrigatório: Sim

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

dryrun

Se definido como `true`, executa somente a validação da solicitação, sem criar nenhum recurso. Use este parâmetro para validar a configuração do cluster e atualizar os requisitos. O padrão é `false`.

Tipo: booliano

Obrigatório: não

forceUpdate

Se definido como `true`, ignora os erros de validação da atualização e força a atualização. O padrão é `false`.

Tipo: booleano

Obrigatório: não

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

suppressValidators

Identifica um ou mais validadores de configuração a serem suprimidos.

Tipo: string

Formato: (ALL | type: [A-Za-z0-9]+)

Obrigatório: não

Exemplo de valores válidos: currentValue, requestedValue, message

validationFailureLevel

O nível mínimo de validação que pode causar falha na atualização.

Tipo: string

Valores válidos: INFO | WARNING | ERROR

Obrigatório: não

Sintaxe da resposta

```
{
  "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 da resposta

changeSet

O conjunto de alterações para a atualização do cluster.

currentValue

O valor atual do parâmetro a ser atualizado.

Tipo: string

parameter

O parâmetro a ser atualizado.

Tipo: string

requestedValue

O valor solicitado para o parâmetro a ser atualizado.

Tipo: string

cluster

cloudformationStackArn

O nome do recurso da Amazon (ARN) da pilha principal do CloudFormation.

Tipo: string

cloudformationStackStatus

Status de pilha do CloudFormation.

Tipo: string

Valores válidos: 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

clusterName

Nome do cluster.

Tipo: string

clusterStatus

Status do cluster.

Tipo: string

Valores válidos: CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE
| DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE |
UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED

region

A Região da AWS em que o cluster foi criado.

Tipo: string

programador

metadados

Os metadados do programador.

name

O nome do programador.

Tipo: string

version

A versão do programador.

Tipo: string

type

O tipo de programador.

Tipo: string

version

Versão do AWS ParallelCluster usada para criar o cluster.

Tipo: string

validationMessages

Uma lista de mensagens com um nível de validação inferior a `validationFailureLevel`. A lista de mensagens é coletada durante a validação da configuração.

id

O ID do validador.

Tipo: string

level

O nível de validação.

Tipo: string

Valores válidos: INFO | WARNING | ERROR

message

A mensagem de validação

Tipo: string

type

O tipo do validador.

Tipo: string

Exemplo

Python

Solicitação

```
$ update_cluster(cluster_name_3x, path/config-file.yaml)
```

Resposta 200

```
{
  '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

Atualiza o status da frota de computação do cluster.

Tópicos

- [Sintaxe da solicitação](#)
- [Corpo da solicitação](#)
- [Sintaxe da resposta](#)
- [Corpo da resposta](#)
- [Exemplo](#)

Sintaxe da solicitação

```
PATCH /v3/clusters/{clusterName}/computeFleet
{
  "status": "string",
  "region": "string"
}
```

Corpo da solicitação

clusterName

O nome do cluster.

Tipo: string

Obrigatório: sim

status

Status da frota de computação.

Tipo: string

Valores válidos: START_REQUESTED | STOP_REQUESTED | ENABLED | DISABLED

Obrigatório: Sim

region

A Região da AWS em que o cluster está.

Tipo: string

Obrigatório: não

Sintaxe da resposta

```
{
  "status": "START_REQUESTED",
  "lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

Corpo da resposta

status

Status da frota de computação.

Tipo: string

Valores válidos: START_REQUESTED | STARTING | RUNNING | PROTECTED | STOP_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

lastStatusUpdateTime

A marca de data e hora que representa a hora da última atualização de status.

Tipo: datetime

Exemplo

Python

Solicitação

```
$ update_compute_fleet(cluster_name_3x, "START_REQUESTED")
```


Resposta 200

```
{
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
tzinfo=tzlocal()),
  'status': 'START_REQUESTED'
}
```

API da biblioteca Python do AWS ParallelCluster

A partir do AWS ParallelCluster versão 3.5.0, você pode acessar o AWS ParallelCluster com a biblioteca Python do AWS ParallelCluster. Você pode acessar a biblioteca AWS ParallelCluster em seu ambiente `pcluster` ou a partir de um runtime AWS Lambda. Saiba como acessar a API AWS ParallelCluster usando a biblioteca Python do AWS ParallelCluster. A biblioteca Python do AWS ParallelCluster oferece a mesma funcionalidade que a API do AWS ParallelCluster.

As operações e os parâmetros da biblioteca Python do AWS ParallelCluster refletem os parâmetros da API quando convertidos para `snake_case` sem letras maiúsculas.

Tópicos

- [Autorização da biblioteca Python do AWS ParallelCluster](#)
- [Como instalar a biblioteca Python do AWS ParallelCluster](#)
- [Operações de API do cluster](#)
- [Operações da API da frota de computação](#)
- [Operações de cluster e log de pilha](#)
- [Operações da API Image](#)
- [Operações de log de imagens e pilhas](#)
- [Exemplo](#)
- [AWS Lambda para a biblioteca Python do AWS ParallelCluster](#)

Autorização da biblioteca Python do AWS ParallelCluster

Especifique as credenciais usando qualquer uma das formas padrão válidas para boto3. Para obter mais informações, consulte a [documentação do boto3](#).

Como instalar a biblioteca Python do AWS ParallelCluster

1. Instale a CLI `pcluster` versão 3.5.0 ou posterior seguindo as instruções fornecidas em [Configuração do AWS ParallelCluster](#).
2. Importe o módulo `pcluster` e comece a usar a biblioteca, conforme mostrado no exemplo a seguir:

```
import pcluster.lib as pc
pc.create_cluster(cluster_name="mycluster", cluster_configuration="config.yaml")
```

Operações de API do cluster

Tópicos

- [list_clusters](#)
- [create_cluster](#)
- [delete_cluster](#)
- [describe_cluster](#)
- [update_cluster](#)

list_clusters

```
list_clusters(region, next_token, cluster_status)
```

Retorna uma lista de clusters existentes.

Parâmetros:

region

Lista os clusters implantados em um determinado Região da AWS.

next_token

O token a ser usado para solicitações paginadas.

cluster_status

Filtros por status do cluster. O padrão é todos os clusters.

Valores válidos: 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)
```

Cria um cluster em uma determinada região.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

cluster_configuration (obrigatório)

A configuração do cluster como um tipo de dados Python.

region

A Região da AWS do cluster.

suppress_validators

Identifica um ou mais validadores de configuração a serem suprimidos.

Formato: (ALL | type:[A-Za-z0-9]+)

validation_failure_level

O nível mínimo de validação que faz com que a criação de cluster falhe. O padrão é ERROR.

Valores válidos: INFO | WARNING | ERROR.

dry_run

Executa a validação da solicitação, sem criar recursos. Use esse parâmetro para validar a configuração do cluster. O padrão é False.

rollback_on_failure

Se definido como True, AWS ParallelCluster inicia automaticamente uma reversão da pilha de clusters em caso de falhas. O padrão é True.

wait

Se definido como `True`, AWS ParallelCluster aguarda a conclusão da operação. O padrão é `False`.

delete_cluster

```
delete_cluster(cluster_name, region, wait)
```

Exclui um cluster em uma determinada região.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

A Região da AWS do cluster.

wait

Se definido como `True`, aguarda a conclusão da operação. O padrão é `False`.

describe_cluster

```
describe_cluster(cluster_name, region)
```

Obtém informações detalhadas sobre um cluster existente.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

A Região da AWS do cluster.

update_cluster

```
update_cluster(cluster_name, cluster_configuration, suppress_validators,  
validation_failure_level, region, force_update, dry_run, wait)
```

Atualiza um cluster em uma determinada região.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

cluster_configuration (obrigatório)

A configuração do cluster como um tipo de dados Python.

suppress_validators

Identifica um ou mais validadores de configuração a serem suprimidos.

Formato: (ALL | type:[A-Za-z0-9]+)

validation_failure_level

O nível mínimo de validação que faz com que a atualização do cluster falhe. O padrão é ERROR.

Valores válidos: INFO | WARNING | ERROR

region

A Região da AWS do cluster.

dry_run

Executa a validação da solicitação sem criar ou atualizar recursos. Use esse parâmetro para validar a configuração do cluster. O padrão é False.

force_update

Se definido como True, força a atualização ignorando os erros de validação da atualização. O padrão é False.

wait

Se definido como True, aguarda a conclusão da operação. O padrão é False.

Operações da API da frota de computação

Tópicos

- [describe_compute_fleet](#)
- [update_compute_fleet](#)
- [delete_cluster_instances](#)
- [describe_cluster_instances](#)

describe_compute_fleet

```
describe_compute_fleet(cluster_name, region)
```

Descreve o status de uma frota computacional de cluster para um determinado cluster.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

Descreve o status da frota de computação de um cluster implantado em um determinado Região da AWS.

update_compute_fleet

```
update_compute_fleet(cluster_name, status, region)
```

Atualiza o status da frota de computação do cluster.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

status (obrigatório)

O status para o qual atualizar.

Valores válidos: START_REQUESTED | STOP_REQUESTED | ENABLED | DISABLED

region

A Região da AWS do cluster.

delete_cluster_instances

```
delete_cluster_instances(cluster_name, region, force)
```

Exclui um cluster em uma determinada região.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

A Região da AWS do cluster.

force

Se definido como True, força a exclusão quando o cluster com o `cluster_name` determinado não for encontrado. O padrão é False.

describe_cluster_instances

```
describe_cluster_instances(cluster_name, region, next_token, node_type, queue_name)
```

Descreve as instâncias do cluster.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

A Região da AWS do cluster.

next_token

O token a ser usado para solicitações paginadas.

node_type

Filtra as instâncias por node_type.

Valores válidos: HeadNode | ComputeNode

queue_name

Filtra as instâncias pelo nome da fila.

Operações de cluster e log de pilha

Tópicos

- [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)
```

Lista fluxos de log de um determinado cluster.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

A Região da AWS do cluster.

filters

Filtra os fluxos de log do cluster.

Formato: 'Name=a,Values=1 Name=b,Values=2,3'

Os filtros aceitos são:

code-dns-name

A forma abreviada do nome DNS privado da instância; por exemplo, `ip-10-0-0-101`.

node-type

O tipo de nó.

Valores válidos: `HeadNode`

next_token

O token a ser usado para solicitações paginadas.

get_cluster_log_events

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,
start_from_head, limit, start_time, end_time)
```

Obtém eventos de log para um determinado cluster e fluxo de log.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

log_stream_name (obrigatório)

Nome do fluxo de logs.

region

A Região da AWS do cluster.

next_token

O token a ser usado para solicitações paginadas.

start_from_head

Se definido como `True`, AWS ParallelCluster retornará primeiro os eventos de log mais antigos. Se definido como `False`, ele retorna primeiro os eventos de log mais recentes. O padrão é `False`.

limit

O número máximo de eventos de log retornados. Se você não especificar um valor, o máximo o número de logs que couberem em um tamanho de resposta de 1 MB, com até 10.000 eventos de log.

start_time

O início do intervalo de tempo para eventos de log, expresso no formato ISO 8601; por exemplo, '2021-01-01T20:00:00Z'. Eventos com uma marca de data/hora igual ou posterior a esse horário estão incluídos.

end_time

O final do intervalo de tempo para eventos de log, expresso no formato ISO 8601; por exemplo, '2021-01-01T20:00:00Z'. Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos.

get_cluster_stack_events

```
get_cluster_stack_events(cluster_name, region, next_token)
```

Obtém eventos da pilha de um determinado cluster.

Parâmetros:

cluster_name (obrigatório)

O nome do cluster.

region

A Região da AWS do cluster.

next_token

O token a ser usado para solicitações paginadas.

Operações da API Image

Tópicos

- [list_images](#)

- [build_image](#)
- [delete_image](#)
- [describe_image](#)

list_images

```
list_images(image_status, region, next_token)
```

Recupera a lista de imagens existentes.

Parâmetros:

image_status (obrigatório)

Filtra por status da imagem.

Valores válidos: AVAILABLE | PENDING | FAILED

region

Lista imagens construídas em uma determinada Região da AWS.

next_token

Token a ser usado para solicitações paginadas.

build_image

```
build_image(image_configuration, image_id, suppress_validators,  
validation_failure_level, dry_run, rollback_on_failure, region)
```

Cria uma imagem do AWS ParallelCluster personalizada em uma determinada região.

Parâmetros:

image_configuration (obrigatório)

Configuração da imagem em dados do Python.

image_id (obrigatório)

O ID da imagem.

suppress_validators

Identifica um ou mais validadores de configuração de imagem a serem suprimidos.

Formato: (ALL | type:[A-Za-z0-9]+)

validation_failure_level

O nível mínimo de validação que faz com que a criação de imagem falhe. O padrão é ERROR.

Valores válidos: INFO | WARNING | ERROR

dry_run

Se definido como `True`, AWS ParallelCluster executa a validação da solicitação, sem criar recursos. Use esse parâmetro para validar a configuração da imagem. O padrão é `False`.

rollback_on_failure

Se definido como `True`, AWS ParallelCluster inicia automaticamente uma reversão da pilha de imagens em caso de falhas. O padrão é `False`.

region

A imagem Região da AWS.

delete_image

```
delete_image(image_id, region, force)
```

Exclui uma imagem em uma determinada região.

Parâmetros:

image_id (obrigatório)

O ID da imagem.

region

A imagem Região da AWS.

force

Se definido como `True`, AWS ParallelCluster força a exclusão se as instâncias estiverem usando a AMI ou se a AMI for compartilhada. O padrão é `False`.

describe_image

```
describe_image(image_id, region)
```

Obtém informações detalhadas sobre uma imagem existente.

Parâmetros:

image_id (obrigatório)

O ID da imagem.

region

A imagem Região da AWS.

Operações de log de imagens e pilhas

Tópicos

- [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)
```

Lista fluxos de log para uma imagem.

Parâmetros:

image_id (obrigatório)

O ID da imagem.

region

A imagem Região da AWS.

next_token

O token a ser usado para solicitações paginadas.

get_image_log_events

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head,
                    limit, start_time, end_time)
```

Obtém eventos de log para uma determinada imagem e fluxo de logs.

Parâmetros:

image_id (obrigatório)

O ID da imagem.

log_stream_name (obrigatório)

Nome do fluxo de logs.

region

A imagem Região da AWS.

next_token

O token a ser usado para solicitações paginadas.

start_from_head

Se definido como `True`, AWS ParallelCluster retornará primeiro os eventos de log mais antigos. Se definido como `False`, ele retorna primeiro os eventos de log mais recentes. O padrão é `False`.

limit

O número máximo de eventos de log retornados. Se você não especificar um valor, o máximo o número de logs que couberem em um tamanho de resposta de 1 MB, com até 10.000 eventos de log.

start_time

O início do intervalo de tempo para eventos de log, expresso no formato ISO 8601; por exemplo, `'2021-01-01T20:00:00Z'`. Eventos com uma marca de data/hora igual ou posterior a esse horário estão incluídos.

end_time

O final do intervalo de tempo para eventos de log, expresso no formato ISO 8601; por exemplo, '2021-01-01T20:00:00Z'. Eventos com uma marca de data/hora igual ou posterior a esse horário não estão incluídos.

get_image_stack_events

```
get_image_stack_events(image_id, region, next_token)
```

Obtém eventos de pilha para uma determinada imagem.

Parâmetros:

image_id (obrigatório)

O ID da imagem.

region

A imagem Região da AWS.

next_token

O token a ser usado para solicitações paginadas.

list_official_images

```
list_official_images(region, os, architecture)
```

Recupera a lista de imagens oficiais do AWS ParallelCluster.

Parâmetros:

region

A imagem Região da AWS.

os

Filtra por distribuição do sistema operacional. O padrão é sem filtro.

architecture

Filtra por arquitetura. O padrão é sem filtro.

Exemplo

Tópicos

- [Criar um cluster do](#)

Criar um cluster do

Ao executar o script de exemplo a seguir, com as entradas fornecidas armazenadas em seu ambiente, você cria um cluster. A configuração do cluster é criada como um tipo de dados do Python com base na [documentação de configuração do 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))
```

Resultado:


```
{'cluster': {'cloudformationStackArn': 'arn:aws:cloudformation:us-east-2:123456789012:stack/mycluster/00000000-aaaa-1111-999-000000000000',
             'cloudformationStackStatus': 'CREATE_IN_PROGRESS',
             'clusterName': 'mycluster',
             'clusterStatus': 'CREATE_IN_PROGRESS',
             'region': 'us-east-2',
             'scheduler': {'type': 'slurm'},
             'version': '3.7.0'}}
```

AWS Lambda para a biblioteca Python do AWS ParallelCluster

Você pode implantar uma camada Lambda e um runtime para acessar a biblioteca Python do AWS ParallelCluster. Hospedamos arquivos zip do AWS ParallelCluster que você pode usar inserindo o link para o arquivo zip conforme descrito nas etapas a seguir. O Lambda usa os arquivos zip para preparar o ambiente de execução para dar suporte ao acesso à biblioteca Python. A biblioteca Python do AWS ParallelCluster foi adicionada com o AWS ParallelCluster versão 3.5.0. Você só pode usar a biblioteca para versões 3.5.0 e posteriores.

O URL do arquivo zip hospedado está no formato: `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`

Comece a acessar a biblioteca Python do AWS ParallelCluster com AWS Lambda

Crie uma camada do Lambda

1. Faça login no AWS Management Console e navegue até o console do AWS Lambda.
2. No painel de navegação, escolha Camadas e Criar uma camada.
3. Insira um nome para sua camada e selecione Fazer upload de um arquivo do Amazon S3.
4. Insira o URL do arquivo zip: `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`.
5. Para Arquiteturas compatíveis, escolha a arquitetura x86_64.
6. Para runtimes compatíveis, escolha o runtime do Python 3.9.
7. Escolha Criar.

Use sua camada do Lambda

1. No painel de navegação do console Lambda, selecione Funções e, em seguida, Criar função.

2. Insira um nome para sua função.
3. Para Runtime, escolha o runtime do Python 3.9.
4. Em Arquitetura, escolha a arquitetura x86_64.
5. Escolha Criar função.
6. Depois que a função for criada, escolha Camadas e selecione Adicionar uma camada.
7. Selecione Camadas personalizadas e escolha a camada que você criou nas etapas anteriores.
8. Escolha a versão da camada.
9. Escolha Add (Adicionar).
10. Seu Lambda precisa de permissões para gerenciar clusters criados com o AWS ParallelCluster. Crie uma função do Lambda com as permissões listadas em [Política básica de usuário pcluster do AWS ParallelCluster](#).

Agora você pode acessar AWS ParallelCluster a partir da biblioteca Python, conforme descrito em [API da biblioteca Python do AWS ParallelCluster](#).

Como funciona o AWS ParallelCluster

O AWS ParallelCluster foi criado não apenas como uma maneira de gerenciar clusters, mas como uma referência sobre como usar os serviços da AWS para criar seu ambiente de HPC.

Tópicos

- [AWS ParallelCluster processos](#)
- [Serviços AWS usados por AWS ParallelCluster](#)
- [AWS ParallelCluster Diretórios internos](#)

AWS ParallelCluster processos

Esta seção se aplica aos clusters que são implantados com o Slurm. Quando usado com esse agendador, AWS ParallelCluster gerencia o provisionamento e a remoção de nós de computação interagindo com o agendador de tarefas subjacente.

Para clusters de HPC baseados em AWS Batch, AWS ParallelCluster depende dos recursos fornecidos pelo AWS Batch para o gerenciamento de nós de computação.

clustermgtd

As seguintes tarefas são executadas pelo daemon de gerenciamento de cluster.

- Limpeza de partições inativas
- Gerenciamento de reservas e nós do Slurm associados aos blocos de capacidade (consulte a seção a seguir)
- Gerenciamento de capacidade estática: certifique-se de que a capacidade estática esteja sempre ativa e saudável
- Programador de sincronização com o Amazon EC2.
- Limpeza de instâncias órfãs
- Restaura o status do nó do programador quando ocorre encerramento do Amazon EC2 fora do fluxo de trabalho de suspensão
- Gerenciamento de instâncias não íntegras do Amazon EC2 (falha nas verificações de integridade do Amazon EC2)

- Gerenciamento de eventos de manutenção programados
- Gerenciamento de nós não íntegros do Scheduler (falha nas verificações de integridade do Scheduler)

Gerenciamento de reservas e nós do Slurm associados aos blocos de capacidade

ParallelCluster suporta reservas de capacidade sob demanda (ODCR) e blocos de capacidade para Machine Learning (CB). Ao contrário do ODCR, o CB pode ter um horário de início futuro e é limitado por um horário.

O Clustermgtd pesquisa nós não íntegros em um loop e encerra todas as instâncias do EC2 que estejam inativas, substituindo-as por novas instâncias se forem nós estáticos.

ParallelCluster gerencia os nós estáticos associados aos blocos de capacidade de forma diferente. AWS ParallelCluster cria um cluster mesmo que o CB ainda não esteja ativo, e as instâncias são iniciadas automaticamente quando o CB está ativo.

Os nós do Slurm correspondentes aos recursos computacionais associados aos CBs que ainda não estão ativos são mantidos em manutenção até que a hora de início do CB seja atingida. Os nós do Slurm permanecerão em um estado de reserva/manutenção associado ao usuário administrador do slurm, o que significa que eles podem aceitar trabalhos, mas os trabalhos permanecerão pendentes até que a reserva do Slurm seja removida.

O Clustermgtd criará e excluirá automaticamente as reservas do Slurm, colocando os nós CB relacionados em manutenção com base no estado do CB. Quando o CB estiver ativo, a reserva do Slurm será removida, os nós serão iniciados e ficarão disponíveis para os trabalhos pendentes ou para o envio de novos trabalhos.

Quando o horário de término do CB for atingido, os nós serão movidos de volta ao estado de reserva/manutenção. Cabe aos usuários reenviar/reenfileirar os trabalhos para uma nova fila/recurso de computação quando o CB não estiver mais ativo e as instâncias forem encerradas.

clusterstatusmgtd

O daemon de gerenciamento de status do cluster gerencia a atualização do status da frota de computação. A cada minuto, ele busca o status da frota armazenado em uma tabela do DynamoDB e gerencia qualquer solicitação STOP/START.

computemgtd

Os processos daemon de gerenciamento de computação (computemgtd) são executados em cada um dos nós de computação do cluster. A cada cinco (5) minutos, o daemon de gerenciamento de computação confirma que o nó principal pode ser alcançado e está íntegro. Se passarem cinco (5) minutos durante os quais o nó principal não puder ser alcançado ou não estiver íntegro, o nó de computação será encerrado.

Serviços AWS usados por AWS ParallelCluster

Os serviços da Amazon Web Services (AWS) a seguir são usados no AWS ParallelCluster.

Tópicos

- [Amazon API Gateway](#)
- [AWS Batch](#)
- [AWS CloudFormation](#)
- [Amazon CloudWatch](#)
- [CloudWatch Eventos da Amazon](#)
- [CloudWatch Registros da Amazon](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx para Lustre](#)
- [Amazon FSx para ONTAP NetApp](#)
- [Amazon FSx para 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](#)
- [NICE DCV](#)

Amazon API Gateway

O Amazon API Gateway é um AWS serviço para criar, publicar, manter, monitorar e proteger REST, HTTP e WebSocket APIs em qualquer escala

O AWS ParallelCluster usa o API Gateway para hospedar a API do AWS ParallelCluster.

Para obter mais informações sobre AWS Batch, consulte <https://aws.amazon.com/api-gateway/> e <https://docs.aws.amazon.com/apigateway/>.

AWS Batch

AWS Batch é um serviço AWS gerenciado de agendamento de tarefas. Ele provisiona dinamicamente a quantidade ideal e o tipo de recursos de computação (por exemplo, CPU ou instâncias otimizadas para memória) nos clusters AWS Batch. Esses recursos são provisionados com base nos requisitos específicos de seus trabalhos em lotes, incluindo os requisitos de volume. Com o AWS Batch, não é necessário instalar e gerenciar o software de computação em lote ou clusters para gerenciar suas tarefas de modo eficaz.

O AWS Batch é usado somente com clusters do AWS Batch.

Para obter mais informações sobre AWS Batch, consulte <https://aws.amazon.com/batch/> e <https://docs.aws.amazon.com/batch/>.

AWS CloudFormation

AWS CloudFormation é um infrastructure-as-code serviço que fornece uma linguagem comum para modelar AWS e provisionar recursos de aplicativos de terceiros em seu ambiente de nuvem. É o principal serviço usado pelo AWS ParallelCluster. Cada cluster em AWS ParallelCluster é representado como uma pilha, e todos os recursos exigidos por cada cluster são definidos no AWS ParallelCluster AWS CloudFormation modelo. Na maioria dos casos, os comandos da AWS ParallelCluster CLI correspondem diretamente aos comandos de AWS CloudFormation pilha, como

os comandos `create`, `update` e `delete`. As instâncias que são executadas em um cluster fazem chamadas de HTTPS para o endpoint do AWS CloudFormation para a Região da AWS na qual o cluster é executado.

Para obter mais informações sobre AWS CloudFormation, consulte <https://aws.amazon.com/cloudformation/> e <https://docs.aws.amazon.com/cloudformation/>.

Amazon CloudWatch

O Amazon CloudWatch (CloudWatch) é um serviço de monitoramento e observabilidade que fornece dados e insights acionáveis. Esses insights podem ser usados para monitorar seus aplicativos, responder a mudanças de desempenho e exceções de serviço e otimizar a utilização de recursos. InAWS ParallelCluster, CloudWatch é usado para um painel, para monitorar e registrar as etapas de criação da imagem do Docker e a saída dos AWS Batch trabalhos.

Antes da AWS ParallelCluster versão 2.10.0, CloudWatch era usado somente com AWS Batch clusters.

Para obter mais informações sobre CloudWatch, consulte <https://aws.amazon.com/cloudwatch/> e <https://docs.aws.amazon.com/cloudwatch/>.

CloudWatch Eventos da Amazon

O Amazon CloudWatch CloudWatch Events (Events) fornece um fluxo quase em tempo real de eventos do sistema que descrevem mudanças nos recursos da Amazon Web Services (AWS). Com regras simples que você pode configurar rapidamente, é possível corresponder eventos e roteá-los para um ou mais streams ou funções de destino. EmAWS ParallelCluster, CloudWatch Events é usado para AWS Batch trabalhos.

Para obter mais informações sobre CloudWatch eventos, consulte <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cwe-now-eb>.

CloudWatch Registros da Amazon

O Amazon CloudWatch CloudWatch Logs (Logs) é um dos principais recursos da Amazon CloudWatch. Você pode usá-lo para monitorar, armazenar, exibir e pesquisar os arquivos de log de diversos componentes usados pelo AWS ParallelCluster.

Antes da AWS ParallelCluster versão 2.6.0, o CloudWatch Logs era usado apenas com AWS Batch clusters.

Para obter mais informações, consulte [Integração com Amazon CloudWatch Logs](#).

AWS CodeBuild

AWS CodeBuild(CodeBuild) é um serviço AWS gerenciado de integração contínua que cumpre o código-fonte, executa testes e produz pacotes de software prontos para implantação. InAWS ParallelCluster, CodeBuild é usado para criar imagens do Docker de forma automática e transparente quando os clusters são criados.

CodeBuild é usado somente com AWS Batch clusters.

Para obter mais informações sobre CodeBuild, consulte <https://aws.amazon.com/codebuild/> e <https://docs.aws.amazon.com/codebuild/>.

Amazon DynamoDB

O Amazon DynamoDB (DynamoDB) é um serviço de banco de dados NoSQL rápido e flexível. É usado para armazenar o estado mínimo de informações do cluster. O nó principal rastreia as instâncias provisionadas em uma tabela do DynamoDB.

O DynamoDB não é usado com clusters do AWS Batch.

Para obter mais informações sobre o DynamoDB, consulte <https://aws.amazon.com/dynamodb/> e <https://docs.aws.amazon.com/dynamodb/>.

Amazon Elastic Block Store

O Amazon Elastic Block Store (Amazon EBS) é um serviço de armazenamento em bloco de alto desempenho que oferece armazenamento persistente para volumes compartilhados. Todas as configurações do Amazon EBS podem ser passadas pela configuração. Os volumes do Amazon EBS podem ser inicializados vazios ou a partir de um snapshot existente do Amazon EBS.

Para obter mais informações sobre o Amazon EBS, consulte <https://aws.amazon.com/ebs/> e <https://docs.aws.amazon.com/ebs/>.

Amazon Elastic Compute Cloud

O Amazon Elastic Compute Cloud (Amazon EC2) oferece uma capacidade de computação para AWS ParallelCluster. A cabeça e os nós de computação são instâncias do Amazon EC2. Qualquer tipo de instância que ofereça suporte ao HVM pode ser selecionada. Os nós principais e de

computação podem ser tipos de instância diferentes. Além disso, se várias filas forem usadas, alguns ou todos os nós de computação também poderão ser executados como uma instância spot. Os volumes de armazenamento de instâncias encontrados nas instâncias são montados como volumes distribuídos do LVM.

Para obter mais informações sobre o Amazon EC2, consulte <https://aws.amazon.com/ebs/> e <https://docs.aws.amazon.com/ebs/>.

Amazon Elastic Container Registry

O Amazon Elastic Container Registry (Amazon ECR) é um registro de contêiner do Docker totalmente gerenciado que facilita o armazenamento, o gerenciamento e a implantação de imagens de contêiner do Docker. No AWS ParallelCluster, o Amazon ECR armazena as imagens do Docker que são criadas quando os clusters são criados. As imagens do Docker são então usadas pelo AWS Batch para executar os contêineres para os trabalhos enviados.

O Amazon ECR é usado somente com clusters do AWS Batch.

Para obter mais informações, consulte <https://aws.amazon.com/ecr/> e <https://docs.aws.amazon.com/ecr/>.

Amazon EFS

O Amazon Elastic File System (Amazon EFS) é um sistema de arquivos NFS simples, escalável e elástico totalmente gerenciado para uso com serviços e recursos on-premises da Nuvem AWS. O Amazon EFS é usado quando [EfsSettings](#) são especificados. O suporte para o Amazon EFS foi adicionado no AWS ParallelCluster versão 2.1.0.

Para obter mais informações sobre o Amazon EFS, consulte <https://aws.amazon.com/efs/> e <https://docs.aws.amazon.com/efs/>.

Amazon FSx para Lustre

O FSx for Lustre fornece um sistema de arquivos de alto desempenho que usa o sistema de arquivos Lustre de código aberto. FSx for Lustre é usado quando [Propriedades do FsxLustreSettings](#) são especificados. O suporte para FSx for Lustre foi adicionado no AWS ParallelCluster versão 2.2.1.

Para obter mais informações sobre o FSx for Lustre, consulte <https://aws.amazon.com/fsx/lustre/> e <https://docs.aws.amazon.com/fsx/>.

Amazon FSx para ONTAP NetApp

O FSx for ONTAP fornece um sistema de armazenamento compartilhado totalmente gerenciado baseado no popular sistema NetApp de arquivos ONTAP. FSx for ONTAP é usado quando [Propriedades do FsxOntapSettings](#) são especificados. O suporte para FSx for ONTAP foi adicionado no AWS ParallelCluster versão 3.2.0.

Para obter mais informações sobre o FSx for ONTAP, consulte <https://aws.amazon.com/fsx/netapp-ontap/> e <https://docs.aws.amazon.com/fsx/>.

Amazon FSx para OpenZFS

O FSx for OpenZFS fornece um sistema de armazenamento compartilhado totalmente gerenciado baseado no popular sistema de arquivos OpenZFS. O FSx for OpenZFS é usado quando [Propriedades do FsxOpenZfsSettings](#) são especificados. O suporte para o FSx for OpenZFS foi adicionado no AWS ParallelCluster versão 3.2.0.

Para obter mais informações sobre o FSx for OpenZFS, consulte <https://aws.amazon.com/fsx/openzfs/> e <https://docs.aws.amazon.com/fsx/>.

AWS Identity and Access Management

O AWS Identity and Access Management (IAM) é usado no AWS ParallelCluster para fornecer um perfil do IAM de privilégio mínimo para o Amazon EC2 para a instância específica para cada cluster. As instâncias do AWS ParallelCluster recebem acesso apenas às chamadas de API específicas que são necessárias para implantar e gerenciar o cluster.

Com clusters do AWS Batch, os perfis do IAM também são criados para os componentes que estão envolvidos com o processo de criação da imagem do Docker no momento de criação do cluster. Esses componentes incluem as funções Lambda que têm permissão para adicionar e excluir imagens do Docker de e para o repositório Amazon ECR. Eles também incluem as funções permitidas para excluir o bucket do Amazon S3 criado para o cluster e CodeBuild o projeto. Também há funções para recursos, instâncias e tarefas do AWS Batch.

Para obter mais informações sobre o IAM, consulte <https://aws.amazon.com/iam/> e <https://docs.aws.amazon.com/iam/>.

AWS Lambda

O AWS Lambda (Lambda) executa as funções que orquestram a criação da imagem do Docker. O Lambda também gerencia a limpeza de recursos de cluster personalizado, como imagens do Docker armazenadas no repositório do Amazon ECR e no Amazon S3.

Para obter mais informações sobre o Lambda, consulte <https://aws.amazon.com/lambda/> e <https://docs.aws.amazon.com/lambda/>.

Amazon RDS

O Amazon Relational Database Service (Amazon RDS) é um serviço da Web que facilita a configuração, operação e escala de um banco de dados relacional na nuvem do AWS.

O AWS ParallelCluster usa o Amazon RDS para AWS Batch e Slurm.

Para obter mais informações sobre o Amazon RDS, consulte <https://aws.amazon.com/rds/> e <https://docs.aws.amazon.com/rds/>.

Amazon Route 53

O Amazon Route 53 (Route 53) é usado para criar zonas hospedadas com nomes de host e nomes de domínio totalmente qualificados para cada um dos nós computacionais.

Para obter mais informações sobre o Route 53, consulte <https://aws.amazon.com/route53/> e <https://docs.aws.amazon.com/route53/>.

Amazon Simple Notification Service

(Amazon SNS) é um serviço gerenciado que oferece entrega de mensagens de publicadores para assinantes (também conhecidos como produtores e consumidores).

O AWS ParallelCluster usa o Amazon SNS para hospedagem de API.

Para obter mais informações sobre o Amazon SNS, consulte <https://aws.amazon.com/sns/> e <https://docs.aws.amazon.com/sns/>.

Amazon Simple Storage Service

O Amazon Simple Storage Service (Amazon S3) armazena modelos do AWS ParallelCluster localizados em cada Região da AWS. O AWS ParallelCluster pode ser configurado para permitir que as ferramentas CLI/SDK usem o Amazon S3.

O AWS ParallelCluster também cria um bucket Amazon S3 em sua Conta da AWS para armazenar recursos que são usados por seus clusters, como o arquivo de configuração do cluster. O AWS ParallelCluster mantém um bucket do Amazon S3 em cada Região da AWS que você criar clusters.

Quando você usa o AWS Batch cluster, um bucket do Amazon S3 em sua conta é usado para armazenar dados relacionados. Por exemplo, o bucket armazena os artefatos usados pela criação de imagens do Docker e scripts de tarefas enviadas.

Para obter mais informações, consulte <https://aws.amazon.com/s3/> e <https://docs.aws.amazon.com/s3/>.

Amazon VPC

O Amazon VPC define uma rede usada pelos nós em seu cluster.

Para obter mais informações sobre o Amazon VPC, consulte <https://aws.amazon.com/vpc/> e <https://docs.aws.amazon.com/vpc/>.

Elastic Fabric Adapter

O Elastic Fabric Adapter (EFA) é uma interface de rede para instâncias do Amazon EC2 que permite que você execute aplicações que exigem altos níveis de comunicações entre nós em escala na AWS.

Para obter mais informações sobre o EC2 Image Builder, consulte <https://aws.amazon.com/hpc/efa/>.

EC2 Image Builder

O EC2 Image Builder é um serviço totalmente AWS gerenciado que ajuda você a automatizar a criação, o gerenciamento e a implantação de imagens personalizadas, seguras up-to-date e de servidor.

O AWS ParallelCluster usa o Image Builder para criar e gerenciar imagens do AWS ParallelCluster.

Para obter mais informações sobre o EC2 Image Builder, consulte <https://aws.amazon.com/image-builder/> e <https://docs.aws.amazon.com/imagebuilder/>.

NICE DCV

O NICE DCV é um protocolo de exibição remota de alto desempenho que fornece uma maneira segura de fornecer desktops remotos e streaming de aplicativos para qualquer dispositivo em

condições de rede variáveis. O NICE DCV é usado quando as configurações [Seção HeadNode / Dcv](#) são especificadas. O suporte para o NICE DCV foi adicionado no AWS ParallelCluster versão 2.5.0.

Para obter mais informações sobre o NICE DCV, consulte <https://aws.amazon.com/hpc/dcv/> e <https://docs.aws.amazon.com/dcv/>.

AWS ParallelClusterDiretórios internos

Há vários diretórios internos AWS ParallelCluster usados para compartilhar dados dentro do cluster. Os seguintes diretórios são compartilhados entre o nó principal, os nós de computação e os nós de login:

`/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

Por padrão, esses diretórios são criados no volume do EBS dos nós principais e compartilhados como exportações de NFS para os nós de computação e login. A partir da AWS ParallelCluster versão 3.8, você pode AWS ParallelCluster habilitar a criação e o gerenciamento de um sistema de arquivos Amazon EFS para hospedar e compartilhar esses diretórios definindo o parâmetro como `efs`. [SharedStorageType](#)

Quando o cluster se expande, as exportações de NFS por meio do volume do EBS podem causar gargalos de desempenho. Usando o EFS, você pode evitar exportações de NFS à medida que seu cluster se expande e evitar gargalos de desempenho associados a elas.

Tutoriais

Os tutoriais a seguir mostram como começar a usar o AWS ParallelCluster versão 3 e oferecem orientação sobre as melhores práticas para algumas tarefas comuns.

Ao usar a interface de linha de AWS ParallelCluster comando (CLI) ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Tópicos

- [Como executar o seu primeiro trabalho no AWS ParallelCluster](#)
- [Criar uma AMI personalizada do AWS ParallelCluster](#)
- [Integração do Active Directory](#)
- [Configurando a criptografia de armazenamento compartilhado com uma chave AWS KMS](#)
- [Executando trabalhos em um cluster de modo de várias filas](#)
- [Uso da API AWS ParallelCluster](#)
- [Criação de um cluster com contabilidade Slurm](#)
- [Revertendo para uma versão anterior do documento do AWS Systems Manager](#)
- [Criação de um cluster com AWS CloudFormation](#)
- [AWS ParallelClusterIntegração da interface do usuário com o Identity Center](#)

Como executar o seu primeiro trabalho no AWS ParallelCluster

Este tutorial orienta você durante a execução de sua primeira tarefa Hello World no AWS ParallelCluster.

Ao usar a interface de linha de AWS ParallelCluster comando (CLI) ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Pré-requisitos

- AWS ParallelCluster [está instalado](#).
- O AWS CLI [está instalado e configurado](#).
- Você tem um [par de chaves EC2](#).
- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.

Verificar a instalação

Primeiro, verificamos se o AWS ParallelCluster está instalado e configurado corretamente, incluindo a dependência Node.js.

```
$ node --version
v16.8.0
$ pcluster version
{
  "version": "3.7.0"
}
```

Isso retorna a versão do AWS ParallelCluster em execução.

Criação de seu primeiro cluster

Agora é hora de criar seu primeiro cluster. Como a carga de trabalho desse tutorial não é de desempenho intensivo, podemos usar o tamanho de instância padrão de `t2.micro`. (Para cargas de trabalho de produção, escolha um tamanho de instância que melhor atenda às suas necessidades.) Vamos chamar seu cluster `hello-world`.

```
$ pcluster create-cluster \
  --cluster-name hello-world \
  --cluster-configuration hello-world.yaml
```

Note

O Região da AWS a ser usado deve ser especificado para a maioria dos comandos `pcluster`. Se não estiver especificado na variável de ambiente `AWS_DEFAULT_REGION` ou na configuração `region` na seção `[default]` do arquivo `~/.aws/config` o parâmetro `--region` deverá ser fornecido na linha de comando `pcluster`.

Se a saída fornecer uma mensagem sobre configuração, será necessário executar o seguinte para configurar o AWS ParallelCluster:

```
$ pcluster configure --config hello-world.yaml
```

Se o comando [pcluster create-cluster](#) for bem-sucedido, você verá um resultado semelhante a este:

```
{
  "cluster": {
    "clusterName": "hello-world",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:xxx:stack/xxx",
    "region": "...",
    "version": "...",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

Você monitora a criação do cluster usando:

```
$ pcluster describe-cluster --cluster-name hello-world
```

Os relatórios `clusterStatus` `"CREATE_IN_PROGRESS"` enquanto o cluster está sendo criado. A `clusterStatus` muda para `"CREATE_COMPLETE"` quando o cluster é criado com sucesso. O resultado também nos fornece `publicIpAddress` e `privateIpAddress` do nosso nó de cabeçalho.

Fazendo login em seu nó de cabeçalho.

Use o arquivo pem OpenSSH para fazer login no nó de cabeçalho.


```
$ pcluster ssh --cluster-name hello-world -i /path/to/keyfile.pem
```

Depois de fazer login, execute o comando `sinfo` para verificar se os nós de computação estão definidos e configurados.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite   10    idle~ queue1-dy-queue1t2micro-[1-10]
```

A saída mostra que temos uma fila em nosso cluster, com até dez nós.

Executar o primeiro trabalho usando o Slurm

Depois, criamos uma tarefa que permanece em espera por um tempo e então emite seu próprio nome de host como saída. Crie um arquivo chamado `hellojob.sh` com o seguinte conteúdo:

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

Depois, envie a tarefa usando `sbatch` e verifique se ela é executada.

```
$ sbatch hellojob.sh
Submitted batch job 2
```

Agora, você pode visualizar a fila e verificar o status do trabalho. O provisionamento de uma nova instância do Amazon EC2 é iniciada em segundo plano. Você pode monitorar o status das instâncias do cluster com o comando `squeue`.

```
$ squeue
          JOBID PARTITION    NAME     USER ST       TIME  NODES NODELIST(REASON)
           2      queue1 hellojob ec2-user CF      3:30      1 queue1-dy-
queue1t2micro-1
```

A saída mostra que o trabalho foi enviado a `queue1`. Aguarde 30 segundos para que a tarefa seja concluída e execute `squeue` novamente.

```
$ squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	ODELIST(REASON)
-------	-----------	------	------	----	------	-------	-----------------

Agora que não há trabalhos na fila, podemos verificar a saída em nosso diretório atual.

```
$ 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
```

Na saída, vemos um arquivo out ". Podemos ver a saída do nosso trabalho:

```
$ cat slurm-2.out
Hello World from queue1-dy-queue1t2micro-1
```

A saída também mostra que a tarefa foi executada com êxito na instância queue1-dy-queue1t2micro-1.

No cluster que você acabou de criar, somente o diretório inicial é compartilhado entre todos os nós do cluster.

Para saber mais sobre como criar e usar clusters, consulte [Práticas recomendadas](#).

Se seu aplicativo exigir software, bibliotecas ou dados compartilhados, considere as seguintes opções:

- Crie uma AMI personalizada habilitada AWS ParallelCluster que inclua seu software conforme descrito em [Criar uma AMI personalizada do AWS ParallelCluster](#).
- Use a opção [StorageSettings](#) no arquivo de configuração AWS ParallelCluster para especificar um sistema de arquivos compartilhado e armazenar o software instalado no local de montagem especificado.
- Use [Ações de bootstrap personalizadas](#) para automatizar o procedimento de bootstrap de cada nó do seu cluster.

Criar uma AMI personalizada do AWS ParallelCluster

Ao usar a interface de linha de comando (CLI) AWS ParallelCluster ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria Free Tier AWS na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Important

Se você criar uma AMI personalizada, será necessário repetir as etapas usadas para criar sua AMI personalizada com cada versão nova do AWS ParallelCluster.

Antes de continuar lendo, recomendamos que você analise antes a seção [Ações de bootstrap personalizadas](#). Determine se as modificações que deseja fazer podem ser expressas em scripts e são compatíveis com versões futuras do AWS ParallelCluster.

Embora a criação de uma AMI personalizada em geral não seja ideal, há cenários específicos em que a criação de uma AMI personalizada para o AWS ParallelCluster é necessária. Este tutorial mostra como criar uma AMI personalizada para esses cenários.

Pré-requisitos

- AWS ParallelCluster [está instalado](#).
- O AWS CLI [está instalado e configurado](#).
- Você tem um [par de chaves EC2](#).
- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a CLI [pcluster](#) e compilar imagens.

Como personalizar a AMI do AWS ParallelCluster

Há duas maneiras de criar uma AMI personalizada do AWS ParallelCluster. Um desses dois métodos é criar uma nova AMI usando a CLI AWS ParallelCluster. Outro método exige que você faça modificações manuais para criar uma nova AMI que esteja disponível em seu Conta da AWS.

Criar uma AMI personalizada do AWS ParallelCluster

Se você tiver uma AMI e um software personalizados, poderá aplicar as alterações necessárias no AWS ParallelCluster em cima deles. O AWS ParallelCluster depende do serviço EC2 Image Builder para criar AMIs personalizadas. Para obter mais informações, consulte o [Guia do usuário do Image Builder](#).

Principais pontos:

- O processo leva cerca de 1 hora. Esse tempo pode variar se houver [Build](#) / [Components](#) adicionais a serem instalados no momento da construção.
- A AMI é marcada com as versões dos componentes principais. Isso inclui o kernel, o programador e o driver [EFA](#). Um subconjunto das versões dos componentes também é relatado na descrição da AMI.
- A partir da AWS ParallelCluster versão 3.0.0, um novo conjunto de comandos da CLI pode ser usado para gerenciar o ciclo de vida das imagens. Isso inclui [build-image](#), [list-images](#), [describe-image](#), e [delete-image](#).
- Esse método é repetível. Você pode executá-lo novamente para manter as AMIs atualizadas (por exemplo, atualizações do sistema operacional) e usá-las ao atualizar um cluster existente.

Note

Se você usar esse método na Partição da AWS na China, poderá obter erros de rede. Por exemplo, você pode ver esses erros no comando `pcluster build-image` quando ele baixa pacotes do GitHub ou de um repositório do sistema operacional. Se ela exceder esse limite, recomendamos usar um dos métodos alternativos a seguir:

1. Siga a abordagem [Modificar uma AMI do AWS ParallelCluster](#) que ignora esse comando.
2. Crie a imagem em outra partição e região, como `us-east-1`, e depois armazene-a e restaure-a para movê-la para a região da China. Para mais informações, consulte [Armazenar e restaurar uma AMI usando o S3](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Etapas:

1. Configure as credenciais Conta da AWS de sua conta para que o cliente AWS ParallelCluster possa fazer chamadas para operações da API AWS em seu nome. Para obter uma lista das permissões necessárias, consulte [AWS Identity and Access Management permissões em AWS ParallelCluster](#).
2. Crie um arquivo básico de configuração de imagem de compilação. Para fazer isso, especifique o [InstanceType](#) a ser usado para criar a imagem e o [ParentImage](#). Eles são usados como ponto de partida para criar a AMI. Para obter mais informações sobre parâmetros opcionais de criação, consulte [Configuração de imagem](#).

Build:

InstanceType: *<BUILD_INSTANCE_TYPE>*

ParentImage: *<BASE_AMI_ID>*

3. Use o comando `pcluster build-image` na CLI para criar uma AMI AWS ParallelCluster a partir daquela fornecida como 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": "3.7.0"  
  }  
}
```

Warning

O `pcluster build-image` usa a VPC padrão. Se você excluir a VPC padrão usando a zona de pouso AWS Control Tower ou AWS, a ID da sub-rede deverá ser especificada no arquivo de configuração da imagem. Para obter mais informações, consulte [SubnetId](#).

Para obter uma lista de outros parâmetros, consulte a página de referência do comando [pcluster build-image](#). Os resultados do comando anterior são os seguintes:

- Uma pilha do CloudFormation é criada com base na configuração da imagem. A pilha inclui todos os recursos do EC2 Image Builder necessários para a construção.
- Os recursos criados incluem os componentes oficiais do Image Builder AWS ParallelCluster aos quais os componentes personalizados do Image Builder podem ser adicionados. Para saber como criar componentes personalizados, consulte os [exemplos de AMIs personalizadas](#) no workshop sobre HPC para clientes do setor público.
- O EC2 Image Builder inicia uma instância de compilação, aplica o livro de receitas AWS ParallelCluster, instala a pilha de software AWS ParallelCluster e executa as tarefas de

configuração necessárias. O livro de receitas AWS ParallelCluster é usado para criar e inicializar AWS ParallelCluster.

- A instância é interrompida e uma nova AMI é criada a partir dela.
 - Outra instância é executada na AMI recém-criada. Durante a fase de teste, o EC2 Image Builder executa testes definidos nos componentes do Image Builder.
 - Se a compilação for bem-sucedida, a pilha será excluída. Se a construção falhar, a pilha será retida e estará disponível para inspeção.
4. Você pode monitorar o status do processo de compilação executando o comando a seguir. Depois que a compilação for concluída, você poderá executá-la para recuperar o ID da AMI fornecido na resposta.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION

# BEFORE COMPLETE
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-
delete.s3.amazonaws.com/parallelcluster/3.7.0/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": "3.7.0",
  "cloudformationStackTags": [
    {
      "value": "3.7.0",
      "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/3.7.0/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": "3.7.0"
}
```

5. Para criar o cluster, insira o ID de AMI no campo [CustomAmi](#) na configuração do cluster.

Solução de problemas e monitoramento do processo de criação de AMI

A criação da imagem é concluída em cerca de uma hora. Você pode monitorar o processo executando o comando [pcluster describe-image](#) ou os comandos de recuperação de log.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

O comando [build-image](#) cria uma pilha do CloudFormation com todos os recursos do EC2 necessários para criar a imagem e inicia o processo do EC2 Image Builder.

Depois de executar o comando [build-image](#), é possível recuperar eventos de pilha do CloudFormation usando o [pcluster get-image-stack-events](#). Você pode filtrar os resultados com o parâmetro `--query` para ver os eventos mais recentes. Para obter mais informações sobre filtragem, consulte a saída de [Filtragem AWS CLI](#) no Guia de usuário AWS Command Line Interface.

```
$ 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/3.7.0/1",
  "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/3.7.0\\\", \\\"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"
}
```

Após cerca de 15 minutos, os eventos da pilha aparecem na entrada de eventos de log relacionada à criação do Image Builder. Agora você pode listar fluxos de logs de imagens e monitorar as etapas do Image Builder usando os comandos [pcluster list-image-log-streams](#) e [pcluster get-image-log-events](#).

```
$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
--query 'logStreams[*].logStreamName'

"3.7.0/1"
]
```



```
$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
--log-stream-name 3.7.0/1 --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-ij90-1234abcd5678/3.7.0/1",
      "timestamp": "2022-04-05T22:13:26.741Z"
    },
    {
      "message": "TOE has completed execution successfully",
      "timestamp": "2022-04-05T22:13:26.819Z"
    }
  ]
}
```

Continue verificando o comando [describe-image](#) até ver o status do 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/3.7.0/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": "3.7.0"
}
```

Se você precisar solucionar um problema de criação de AMI personalizada, crie um arquivo dos logs de imagem conforme descrito nas etapas a seguir.

É possível arquivar os logs em um bucket do Amazon S3 ou em um arquivo local, dependendo do parâmetro `--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"
}
```

O arquivo contém os CloudWatch Logs Streams relacionados ao processo do Image Builder e aos eventos da pilha AWS CloudFormation. O comando pode levar alguns minutos para ser executado.

Gerenciando AMIs personalizadas

A partir da AWS ParallelCluster 3.0.0, um novo conjunto de comandos foi adicionado à CLI para criar, monitorar e gerenciar o ciclo de vida da imagem. Para obter mais informações sobre os comandos, consulte [comandos pcluster](#).

Modificar uma AMI do AWS ParallelCluster

Esse método consiste em modificar uma AMI AWS ParallelCluster oficial adicionando personalização a ela. As AMIs AWS ParallelCluster básicas são atualizadas com novas versões. Essas AMIs têm todos os componentes necessários para que o AWS ParallelCluster funcione quando instalado e configurado. Você pode começar com um deles como base.

Principais pontos:

- Esse método é mais rápido que o comando [build-image](#). No entanto, é um processo manual e não pode ser repetido automaticamente.
- Com esse método, você não tem acesso aos comandos de recuperação de logs e gerenciamento do ciclo de vida da imagem que estão disponíveis por meio da CLI.

Etapas:

New EC2 console

1. Encontre a AMI que corresponde à Região da AWS específica que você usa. Para encontrá-la, use o comando [pcluster list-official-images](#) com o parâmetro `--region` para selecionar os parâmetros específicos Região da AWS e `--os` e `--architecture` para filtrar a AMI desejada com o sistema operacional e a arquitetura que você deseja usar. Na saída, recupere a ID da imagem do EC2.
2. Faça login no AWS Management Console e abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
3. No painel de navegação, escolha **Imagens (Imagens)**, e depois **AMIs**. Pesquise a ID de imagem do EC2 recuperada, selecione a AMI e escolha **Iniciar instância a partir da AMI**.
4. Role para baixo e escolha seu tipo de instância.
5. Escolha seu par de chaves e execute a instância.
6. Faça login na instância usando o usuário do SO e chave SSH.
7. Personalize manualmente sua instância para atender às suas necessidades.
8. Execute o seguinte comando para preparar a instância para a criação da AMI.

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. No console, escolha **Instance state (Estado da instância)** e **Stop instance (Interromper instância)**.

Vá para Instâncias, escolha a nova instância, selecione Instance state (Estado da instância), e Stop instance (Interromper instância).

10. Crie uma nova AMI a partir da instância usando o console do EC2 ou AWS CLI [create-image](#).

Do console do EC2.

- a. No painel de navegação, escolha Instances (Instâncias).
- b. Escolha a instância que você criou e modificou.
- c. Em Actions (Ações), escolha Image (Imagem), e, em seguida, Create Image (Criar imagem)..
- d. Escolha Create Image.

11. Insira o ID de AMI no campo [CustomAmi](#) na configuração do cluster e crie um cluster.

Old EC2 console

1. Encontre a AMI AWS ParallelCluster que corresponde à Região da AWS específica que você usa. Para encontrá-la, você pode usar o comando [pcluster list-official-images](#) com o parâmetro `--region` para selecionar os parâmetros específicos Região da AWS e `--os` e `--architecture` para filtrar a AMI desejada com o sistema operacional e a arquitetura que você deseja usar. Na saída, você pode recuperar o ID da imagem do EC2.
2. Faça login no AWS Management Console e abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
3. No painel de navegação, escolha Images (Imagens), e depois AMIs. Defina o filtro para Imagens públicas e pesquise a ID de imagem do EC2 recuperada, selecione a AMI e escolha Executar.
4. Escolha seu tipo de instância e selecione Next: Configure Instance Details (Avançar: Configurar detalhes da instância) ou Review and Launch (Revisar e iniciar) para iniciar sua instância.
5. Escolha Executar, selecione seu par de chaves, e Executar instâncias.
6. Faça login na instância usando o usuário do SO e chave SSH. Para obter mais informações, navegue até Instâncias, selecione a nova instância e Connect (Conectar).
7. Personalize manualmente sua instância para atender às suas necessidades.
8. Execute o seguinte comando para preparar a instância para a criação da AMI:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. No console do EC2, escolha Instances (instâncias) no painel de navegação, selecione sua nova instância e escolha Actions (Ações), Instance State (Estado da instância) e Stop (Interromper)
10. Crie uma nova AMI a partir da instância usando o console do EC2 ou AWS CLI [create-image](#).

Do console do EC2.

- a. No painel de navegação, escolha Instances (Instâncias).
 - b. Escolha a instância que você criou e modificou.
 - c. Em Actions (Ações), escolha Image (Imagem), e, em seguida, Create Image (Criar imagem).
 - d. Escolha Create Image.
11. Insira o ID de AMI no campo [CustomAmi](#) na configuração do cluster e crie um cluster.

Integração do Active Directory

Neste tutorial, você criará um ambiente de vários usuários. Esse ambiente inclui um AWS ParallelCluster que está integrado a um AWS Managed Microsoft AD (Active Directory) em `corp.example.com`. Você configura um usuário Admin para gerenciar o diretório, um usuário ReadOnly para ler o diretório e um usuário `user000` para fazer login no cluster. Você pode usar o caminho automatizado ou o caminho manual para criar os recursos de rede, um Active Directory (AD) e a instância do EC2 que você usa para configurar o AD. Independentemente do caminho, a infraestrutura que você cria é pré-configurada para integrar AWS ParallelCluster usando um dos seguintes métodos:

- LDAPS com verificação de certificado (recomendado como a opção mais segura)
- LDAPS sem verificação de certificado
- LDAP

O LDAP por si só não fornece criptografia. Para garantir a transmissão segura de informações potencialmente confidenciais, é altamente recomendável que você use LDAPS (LDAP sobre TLS/SSL) para clusters integrados com ADs. Para obter mais informações, consulte [Habilitar o LDAPS do lado do servidor usando AWS Managed Microsoft AD](#) no Guia de Administração AWS Directory Service.

Depois de criar esses recursos, continue configurando e criando seu cluster integrado ao Active Directory (AD). Após criar o cluster, faça login como o usuário que você criou. Para obter mais informações sobre a configuração que você cria neste tutorial, consulte [Acesso de vários usuários aos clusters](#) e a seção de configuração [DirectoryService](#).

Este tutorial aborda como criar um ambiente que ofereça suporte ao acesso de vários usuários aos clusters. Este tutorial não aborda como você cria e usa um AD AWS Directory Service. As etapas que você segue para configurar um AWS Managed Microsoft AD neste tutorial são fornecidas somente para fins de teste. Eles não são fornecidos para substituir a documentação oficial e as melhores práticas que você pode encontrar no [AWS Managed Microsoft AD](#) e [Simple AD](#) no Guia de Administração AWS Directory Service.

Note

As senhas de usuário do diretório expiram de acordo com as definições da propriedade da política de senha do diretório. Para obter mais informações, consulte [Configurações de políticas compatíveis](#). Para redefinir as senhas do diretório com AWS ParallelCluster, consulte [Como redefinir uma senha de usuário e senhas expiradas](#).

Note

Os endereços IP do controlador de domínio do diretório podem mudar devido às alterações do controlador de domínio e à manutenção do diretório. Se você escolheu o método automatizado de criação rápida para criar a infraestrutura de diretórios, deverá alinhar manualmente o balanceador de carga na frente dos controladores de diretório quando os endereços IP do diretório mudarem. Se você usar o método de criação rápida, os endereços IP do diretório não serão alinhados automaticamente com os balanceadores de carga.

Ao usar a interface de linha de comando (CLI) AWS ParallelCluster ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Pré-requisitos

- AWS ParallelCluster [está instalado](#).
- O AWS CLI [está instalado e configurado](#).
- Você tem um [par de chaves EC2](#).
- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.

Ao longo do tutorial, substitua *inputs highlighted in red*, como *region-id* e *d-abcdef01234567890*, por seus próprios nomes e IDs. Substituir *0123456789012* por seu número Conta da AWS.

Etapa 1: Criar a infraestrutura do AD

Escolha a guia Automatizado para criar a infraestrutura do Active Directory (AD) com um modelo de criação AWS CloudFormation rápida.

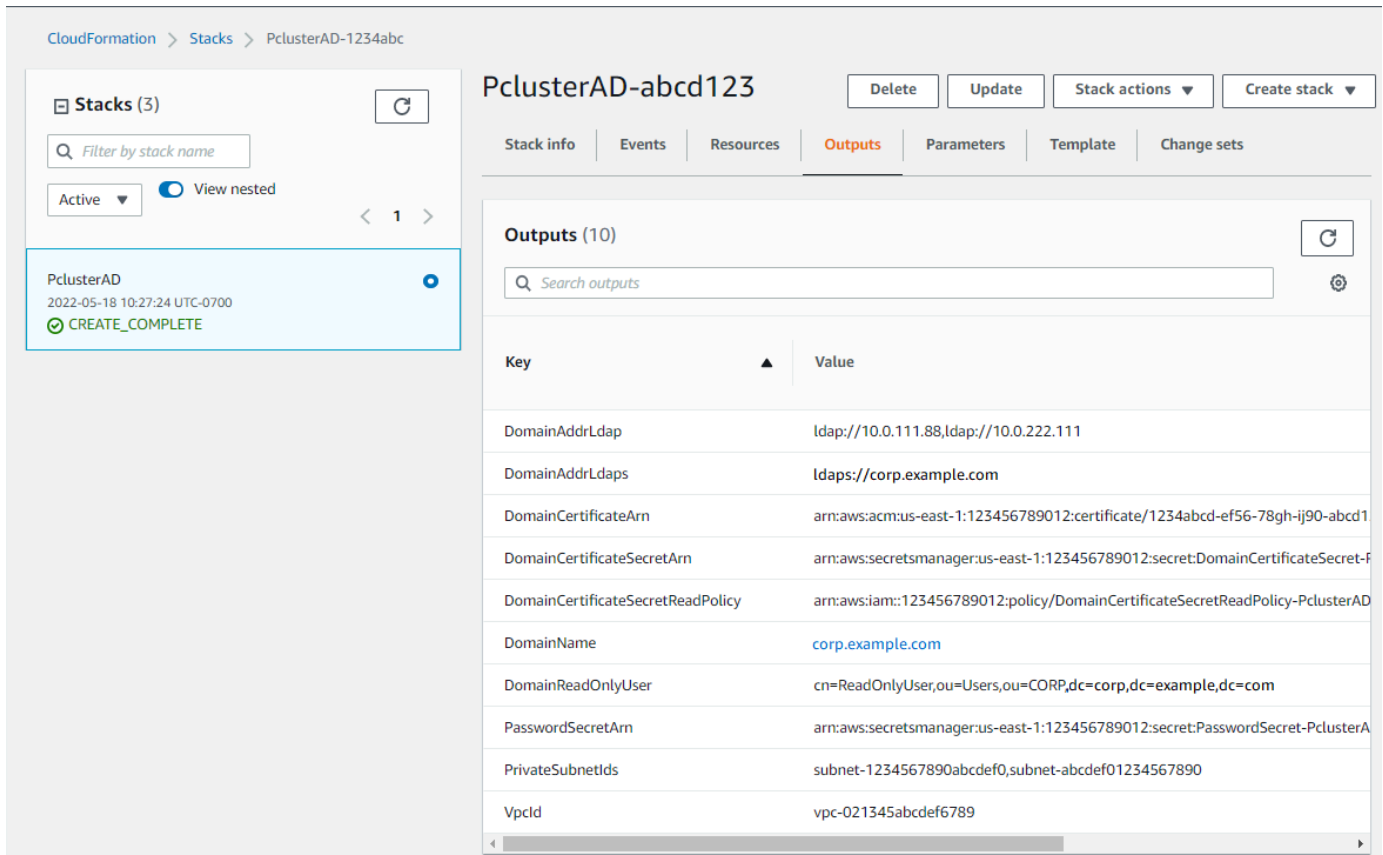
Escolha a guia Manual para criar manualmente a infraestrutura do AD.

Automatizada

1. Faça login no AWS Management Console.
2. Abra a [CloudFormation Criação Rápida \(região us-east-1\)](#) para criar os seguintes recursos no CloudFormation console:
 - Uma VPC com duas sub-redes e roteamento para acesso público, se nenhuma VPC for especificada.
 - Uma AWS Managed Microsoft AD.
 - Uma instância do EC2 associada ao AD que você pode usar para gerenciar o diretório.
3. Na seção Parâmetros da página de pilha de criação rápida, insira valores para os seguintes parâmetros:
 - AdminPassword
 - ReadOnlyPassword
 - UserPassword

Anote a senha. Você precisará dela posteriormente neste tutorial.

4. Em `DomainName`, insira **corp.example.com**
5. Em `Keypair`, insira o nome do par de chaves do EC2.
6. Na parte inferior da página, marque as caixas de seleção que reconhecem cada um dos recursos de acesso.
7. Selecione Criar pilha.
8. Depois que a CloudFormation pilha atingir o `CREATE_COMPLETE` estado, escolha a guia Saídas da pilha. Anote os nomes e IDs dos recursos de saída porque você precisa usá-los em etapas posteriores. As saídas fornecem as informações necessárias para criar o cluster.



The screenshot shows the AWS CloudFormation console for a stack named 'PclusterAD-abcd123'. The stack is in the 'CREATE_COMPLETE' state. The 'Outputs' tab is selected, displaying a table of 10 outputs. The table has columns for 'Key' and 'Value'.

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. Para concluir os exercícios [\(Opcional\) Etapa 2: gerenciar usuários e grupos do AD](#), você precisa do ID do diretório. Escolha Recursos e role para baixo para anotar o ID do diretório.
10. Continue em [\(Opcional\) Etapa 2: gerenciar usuários e grupos do AD](#) ou [Etapa 1: criar um cluster](#).

Manual

Crie uma VPC no serviço de diretório com as duas sub-redes em diferentes zonas de disponibilidade e um AWS Managed Microsoft AD.

Crie o AD

Note

- O diretório e o nome do domínio são `corp.example.com`. O nome curto é `CORP`.
- Altere a senha `Admin` no script.
- O Active Directory (AD) leva pelo menos 15 minutos para ser criado.

Use o script Python a seguir para criar a VPC, as sub-redes e os recursos do AD em sua Região da AWS local. Salve esse arquivo como `ad.py` e execute-o.

```
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})
```

A seguir está um exemplo de saída do script do 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"
}
```

Anote os nomes e IDs dos recursos de saída. Você vai usá-los em etapas subsequentes.

Após concluir o script, avance para a próxima etapa.

Criar uma instância do EC2

New EC2 console

1. Faça login no AWS Management Console.
2. Se você não tiver uma função com as políticas listadas na etapa 4 em anexo, abra o console do IAM em <https://console.aws.amazon.com/iam/>. Caso contrário, vá para a etapa 8.
3. Crie a política ResetUserPassword, substituindo o conteúdo destacado em vermelho por seu ID Região da AWS, ID da conta e ID do diretório da saída do script que você executou para criar o AD.

ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. Crie um perfil do IAM com as políticas a seguir anexada.
 - AWSpolítica gerenciada [AmazonSSM ManagedInstanceCore](#)

- AWSpolítica gerenciada [AmazonSSM DirectoryServiceAccess](#)
 - ResetUserPassword política
5. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
 6. No painel do EC2, escolha Launch Instance (Executar instância).
 7. Em Imagens de aplicações e sistemas operacionais, selecione uma AMI recente do Amazon Linux 2.
 8. Em tipo de instância, escolha t2.micro.
 9. Em par de chaves, escolha um par de chaves.
 10. Em Configurações de rede, escolha Editar.
 11. Em VPC, selecione a VPC do diretório.
 12. Role para baixo e selecione Detalhes avançados.
 13. Em Detalhes avançados, Diretório de ingresso em domínio, escolha **corp.example.com**.
 14. Para o perfil de instância do IAM, escolha a função que você criou na etapa 1 ou uma função com as políticas listadas na etapa 4 em anexo.
 15. Em Resumo, escolha Launch instance (Executar instância).
 16. Anote o ID da instância (por exemplo, i-1234567890abcdef0) e aguarde a conclusão da execução da instância.
 17. Após a instância ser executada, vá para a próxima etapa.

Old EC2 console

1. Faça login no AWS Management Console.
2. Se você não tiver uma função com as políticas listadas na etapa 4 em anexo, abra o console do IAM em <https://console.aws.amazon.com/iam/>. Caso contrário, vá para a etapa 8.
3. Crie a política do ResetUserPassword Substitua o conteúdo destacado em vermelho por seu IDRegião da AWS, ID Conta da AWS, ID do diretório da saída do script que você executou para criar o Active Directory (AD).

ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
```

```
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. Crie um perfil do IAM com as políticas a seguir anexada.
 - AWSpolítica gerenciada [AmazonSSM ManagedInstanceCore](#)
 - AWSpolítica gerenciada [AmazonSSM DirectoryServiceAccess](#)
 - Política do ResetUserPassword
5. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
6. No painel do EC2, escolha Launch Instance (Executar instância).
7. Em Imagens de aplicações e sistemas operacionais, selecione uma AMI recente do Amazon Linux 2.
8. Em Instance type (Tipo de instância), escolha t2.micro.
9. Em par de chaves, escolha um par de chaves.
10. Em Configurações de rede, escolha Editar.
11. Em Network settings (Configurações de rede), VPC, selecione o diretório VPC.
12. Role para baixo e selecione Detalhes avançados.
13. Em Detalhes avançados, Diretório de ingresso em domínio, escolha **corp.example.com**.
14. Em Detalhes avançados, Perfil da instância, escolha a função que você criou na etapa 1 ou uma função com as políticas listadas na etapa 4 em anexo.
15. Em Resumo, escolha Launch instance (Executar instância).
16. Anote o ID da instância (por exemplo, i-1234567890abcdef0) e espere que a instância termine de ser executada.
17. Após a instância ser executada, vá para a próxima etapa.

Associe sua instância ao AD

1. Conecte-se à sua instância e entre no realm do AD como **admin**.

Execute os comandos a seguir para se conectar à instância.

```
$ 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. Instale o software necessário e entre no realm.

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-  
common-tools krb5-workstation openldap-clients policycoreutils-python
```

3. Substitua a senha do administrador pela sua senha **admin**.

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com  
Password for Admin:
```

Se o procedimento anterior tiver sido bem-sucedido, você se juntará ao realm e poderá avançar para a etapa seguinte.

Adicionar usuários ao AD

1. Crie o ReadOnlyUser e um usuário adicional.

Nesta etapa, você usa as ferramentas [adcli](#) e [openldap-clients](#) que você instalou na etapa anterior.

```
$ 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. Verifique se os usuários foram criados:

Os endereços IP DNS do diretório são saídas do 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"
```

Por padrão, quando você cria um usuário com o `ad-cli`, o usuário é desabilitado.

3. Redefina e ative as senhas de usuário da sua máquina local:

Faça logout da instância do EC2.

Note

- `ro-p@ssw0rd` é a senha de `ReadOnlyUser`, recuperada de AWS Secrets Manager.
- `user-p@ssw0rd` é a senha de um usuário do cluster fornecida quando você se conecta (ssh) ao cluster.

`directory-id` é uma saída do 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" \  
--region "region-id"
```

```
--new-password "user-p@ssw0rd" \  
--region "region-id"
```

4. Adicione a senha a um segredo do Secrets Manager.

Agora que você criou `ReadOnlyUser` e definiu a senha, armazene-a em um segredo usado pelo AWS ParallelCluster para validar logins.

Use o Secrets Manager para criar um novo segredo para manter a senha do `ReadOnlyUser` como valor. O formato do valor secreto deve ser somente texto sem formatação (não no formato JSON). Anote o ARN secreto para futuras etapas.

```
$ 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
```

LDAPS com configuração de verificação de certificado (recomendado)

Anote os IDs de recursos. Você vai usá-los em etapas subsequentes.

1. Gere certificado de domínio 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. Armazene o certificado no Secrets Manager para torná-lo recuperável de dentro do cluster posteriormente.

```
$ 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. Adicione a seguinte política ao perfil do IAM que você criou para unir a instância do EC2 ao domínio AD.

PutDomainCertificateSecrets

```
{
  "Statement": [
    {
      "Action": [
        "secretsmanager:PutSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",
      ],
      "Effect": "Allow"
    }
  ]
}
```

4. Importe o certificado para 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. Crie um balanceador de carga que é colocado na frente dos endpoints do 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": "Z2IF0LAFXWL04F",
    "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. Crie o grupo-alvo que tem como alvo os endpoints do 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. Registre os endpoints do Active Directory (AD) no grupo de destino.

```

$ 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. Crie o receptor LB com o certificado.

```

$ 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. Crie a zona hospedada para tornar o domínio detectável dentro da VPC do 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. Crie um arquivo chamado **recordset-change.json** com o conteúdo a seguir. O **HostedZoneId** é o ID de zona hospedada canônica do balanceador de carga.

```

{
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "corp.example.com",
        "Type": "A",
        "Region": "region-id",
        "SetIdentifier": "example-active-directory",
        "AliasTarget": {
          "HostedZoneId": "Z2IFOLAFXWL04F",
          "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}

```

11. Envie a alteração do conjunto de registros para a zona hospedada, desta vez usando o ID da zona hospedada.

```

$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNMSJUB \
--change-batch file://recordset-change.json
{
  "ChangeInfo": {
    "Id": "/change/C0137926I56R3GC7XW2Y",
    "Status": "PENDING",

```

```

    "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
  }
}

```

12. Crie um documento de política **policy.json** com o conteúdo a seguir.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-abc123"
      ],
      "Effect": "Allow"
    }
  ]
}

```

13. Crie um documento de política chamado **policy.json** com o conteúdo a seguir.

```

$ 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. Continue seguindo as etapas em [\(Opcional\) Etapa 2: gerenciar usuários e grupos do AD](#) ou [Etapa 1: criar um cluster](#).

(Opcional) Etapa 2: gerenciar usuários e grupos do AD

Nesta etapa, você gerencia usuários e grupos de uma instância EC2 Amazon Linux 2 que está associada ao domínio Active Directory (AD).

Se você seguiu o caminho automatizado, reinicie e faça login na instância associada ao AD que foi criada como parte da automação.

Se você seguiu o caminho manual, reinicie e faça login na instância que você criou e ingressou no AD nas etapas anteriores.

Nessas etapas, você usa as ferramentas [adcli](#) e [openldap-clients](#) que foram instaladas na instância como parte da etapa anterior.

Faça login em uma instância do EC2 que está associada ao domínio AD

1. No console do EC2, selecione a instância do EC2 sem título que foi criada nas etapas anteriores. A instância deve se encontrar no estado Interrompido.
2. Se o estado da instância for Interrompido, escolha Estado da instância e, em seguida, Iniciar instância.
3. Depois que as verificações de status forem aprovadas, selecione a instância e escolha Conectar e SSH na instância.

Gerencie usuários e grupos quando conectados a uma instância EC2 Amazon Linux 2 que se juntou ao AD

Ao executar os comandos `adcli` com a opção `-U "Admin"`, você será solicitado a inserir a senha Admin do AD. Você inclui a senha Admin do AD como parte dos comandos `ldapsearch`.

1. Criar um usuário.

```
$ adcli create-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

2. Definir uma senha de usuário

```
$ aws --region "region-id" ds reset-user-password --directory-id "d-
abcdef01234567890" --user-name "clusteruser" --new-password "new-p@ssw0rd"
```

3. Criar um grupo

```
$ adcli create-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

4. Adicionar um usuário a um grupo

```
$ adcli add-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

5. Descrever usuários e grupos.

Descrever todos os usuários.

```
$ 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"
```

Descrever um usuário específico.

```
$ 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"
```

Descrever todos os usuários com um padrão de nome.

```
$ 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"
```

Descrever todos os usuários que fazem parte de um grupo específico.

```
$ 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"
```


Descrever todos os grupos

```
$ 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"
```

Descrever um grupo específico

```
$ 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. Remover um usuário de um grupo.

```
$ adcli remove-member "clusterteam" "clusteruser" --domain "corp.example.com" -U "Admin"
```

7. Excluem um usuário.

```
$ adcli delete-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

8. Excluir um grupo

```
$ adcli delete-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

Etapa 1: criar um cluster

Se você ainda não saiu da instância do EC2, faça isso agora.

O ambiente está configurado para criar um cluster que possa autenticar usuários no Active Directory (AD).

Crie uma configuração de cluster simples e forneça as configurações relevantes para se conectar ao AD. Para obter mais informações, consulte a seção [DirectoryService](#).

Escolha uma das seguintes configurações de cluster e copie-a em um arquivo chamado `ldaps_config.yaml`, `ldaps_nocert_config.yaml`, ou `ldap_config.yaml`.

Recomendamos que você escolha a configuração LDAPS com verificação de certificado. Se você escolher essa configuração, também deverá copiar o script de bootstrap em um arquivo chamado

`active-directory.head.post.sh`. E você deve armazená-lo em um bucket do Amazon S3, conforme indicado no arquivo de configuração.

LDAPS com configuração de verificação de certificado (recomendado)

Note

Os componentes a seguir devem ser alterados.

- `KeyName`: Um dos seus pares de chaves do EC2.
- `SubnetId` / `SubnetIds`: um dos IDs de sub-rede fornecidos na saída da pilha de criação CloudFormation rápida (tutorial automatizado) ou do script python (tutorial manual).
- `Region`: A região em que você criou a infraestrutura do AD.
- `DomainAddr`: Esse endereço IP é um dos endereços DNS do seu serviço AD.
- `PasswordSecretArn`: O nome de recurso da Amazon (ARN) do segredo que contém a senha para o `DomainReadOnlyUser`.
- `BucketName`: o nome do bucket que contém o script de bootstrap.
- `AdditionalPolicies/Policy`: O Amazon Resource Name (ARN) da política de certificação de domínio lido. `ReadCertExample`
- `CustomActions / OnNodeConfigured / Args`: O nome de recurso da Amazon (ARN) do segredo que contém a política de certificação de domínio.

Para uma melhor postura de segurança, sugerimos usar a `AllowedIps` configuração `HeadNode/Ssh` para limitar o acesso SSH ao nó principal.

```
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: my-bucket
  EnableWriteAccess: false
  KeyName: bootstrap/active-directory/active-directory.head.post.sh
CustomActions:
  OnNodeConfigured:
    Script: s3://my-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 de bootstrap

Depois de criar o arquivo bootstrap e antes de carregá-lo no bucket do S3, execute `chmod +x active-directory.head.post.sh` para dar ao AWS ParallelCluster permissão de execução.

```

#!/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
```

LDAPS sem configuração de verificação de certificado

Note

Os componentes a seguir devem ser alterados.

- **KeyName:** Um dos seus pares de chaves do EC2.
- **SubnetId / SubnetIds:** um dos IDs de sub-rede que está na saída da pilha de criação CloudFormation rápida (tutorial automatizado) ou do script python (tutorial manual).
- **Region:** A região em que você criou a infraestrutura do AD.
- **DomainAddr:** Esse endereço IP é um dos endereços DNS do seu serviço AD.
- **PasswordSecretArn:** O nome de recurso da Amazon (ARN) do segredo que contém a senha para o DomainReadOnlyUser.

Para uma melhor postura de segurança, sugerimos usar a AllowedIps configuração HeadNode /Ssh/ para limitar o acesso SSH ao nó principal.

```
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
```

Configuração LDAP

Note

Os componentes a seguir devem ser alterados.

- `KeyName`: Um dos seus pares de chaves do EC2.
- `SubnetId` / `SubnetIds`: um dos IDs de sub-rede fornecidos na saída da pilha de criação CloudFormation rápida (tutorial automatizado) ou do script python (tutorial manual).
- `Region`: A região em que você criou a infraestrutura do AD.
- `DomainAddr`: Esse endereço IP é um dos endereços DNS do seu serviço AD.
- `PasswordSecretArn`: O nome de recurso da Amazon (ARN) do segredo que contém a senha para o `DomainReadOnlyUser`.

Para uma melhor postura de segurança, sugerimos usar a `AllowedIps` configuração `HeadNode /Ssh/` para limitar o acesso SSH ao nó principal.

```
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

```

Crie o cluster usando o comando a seguir.

```

$ 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": 3.7.0,
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}

```

Etapa 4: conexão ao cluster enquanto usuário

Você pode determinar o status do cluster com os comandos a seguir.

```
$ pcluster describe-cluster -n ad-cluster --region "region-id" --query "clusterStatus"
```

A saída é a seguinte:

```
"CREATE_IN_PROGRESS" / "CREATE_COMPLETE"
```

Quando o status chegar a "CREATE_COMPLETE", faça login com o nome de usuário e a senha criados.

```
$ HEAD_NODE_IP=$(pcluster describe-cluster -n "ad-cluster" --region "region-id" --query  
headNode.publicIpAddress | xargs echo)
```

```
$ ssh user000@$HEAD_NODE_IP
```

Você pode fazer login sem a senha fornecendo a chave SSH que foi criada para o novo usuário em /home/user000@HEAD_NODE_IP/.ssh/id_rsa.

Se o comando ssh for bem-sucedido, você se conectou com êxito ao cluster como um usuário autenticado para usar o Active Directory (AD).

Etapa 5: limpar

1. Na sua máquina local, exclua o 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": "3.7.0",  
    "clusterStatus": "DELETE_IN_PROGRESS"  
  }  
}
```

2. Verifique o progresso do cluster que está sendo excluído.

```
$ pcluster describe-cluster --cluster-name "ad-cluster" --region "region-id" --  
query "clusterStatus"  
"DELETE_IN_PROGRESS"
```

Após excluir o cluster com sucesso, avance para a próxima etapa.

Automatizada

Excluir os recursos do Active Directory

1. Em <https://console.aws.amazon.com/cloudformation/>.
2. No painel de navegação, escolha Pilhas.
3. Na lista de pilhas, escolha a pilha AD (por exemplo, pcluster-ad).
4. Escolha Excluir.

Manual

1. Excluir a instância do EC2.
 - a. Em <https://console.aws.amazon.com/ec2/>, escolha Instâncias no painel de navegação.
 - b. Na lista de instâncias, selecione a instância que você criou para adicionar usuários ao diretório.
 - c. Escolha Instance state (Estado da instância) e Terminate instance (Terminar instância).
2. Excluir a zona hospedada.
 - a. Crie um `recordset-delete.json`, com o seguinte conteúdo. Neste exemplo, `HostedZoneId` é o ID canônico da zona hospedada do balanceador de carga.

```
{  
  "Changes": [  
    {  
      "Action": "DELETE",  
      "ResourceRecordSet": {  
        "Name": "corp.example.com",  
        "Type": "A",  
        "Region": "region-id",
```



```

    "SetIdentifier": "pcluster-active-directory",
    "AliasTarget": {
      "HostedZoneId": "Z2IFOLAFXWL04F",
      "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
      "EvaluateTargetHealth": true
    }
  }
}
]
}

```

- b. Envie a alteração do conjunto de registros para a zona hospedada usando o ID da zona hospedada.

```

$ 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. Excluir a zona hospedada.

```

$ aws route53 delete-hosted-zone --id Z09020002B5MZQNXSJUB
{
  "ChangeInfo": {
    "Id": "/change/C0468051QFABTVHMDEG9",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T14:26:13.814000+00:00"
  }
}

```

3. Exclui o receptor do LB.

```

$ aws elbv2 delete-listener \
  --listener-arn arn:aws:elasticloadbalancing:region-id:123456789012:listener/net/
CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b --region region-id

```

4. Excluir um grupo de destino.

```
$ aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:region-  
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 --  
region region-id
```

5. Excluir o balanceador de carga.

```
$ aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:region-  
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4 --  
region region-id
```

6. Excluir a política que o cluster usa para ler o certificado do Secrets Manager.

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/  
ReadCertExample
```

7. Excluir o segredo que contém o certificado de domínio.

```
$ 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. Excluir o certificado do ACM.

```
$ aws acm delete-certificate \  
  --certificate-arn arn:aws:acm:region-  
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 --region region-id
```

9. Excluir os recursos do Active Directory (AD).

a. Obtenha os seguintes IDs de recursos da saída do script `python:ad.py`:

- ID AD

- IDs de sub-rede do AD
- ID VPC AD

b. Exclua o diretório executando o comando a seguir.

```
$ aws ds delete-directory --directory-id d-abcdef0123456789 --region region-id
{
  "DirectoryId": "d-abcdef0123456789"
}
```

c. Liste os grupos de segurança na VPC.

```
$ aws ec2 describe-security-groups --filters '[{"Name":"vpc-id","Values":
["vpc-07614ade95ebad1bc"]}]' --region region-id
```

d. Exclua o grupo de segurança personalizado.

```
$ aws ec2 delete-security-group --group-id sg-021345abcdef6789 --region region-id
```

e. Exclua as sub-redes.

```
$ aws ec2 delete-subnet --subnet-id subnet-1234567890abcdef --region region-id
```

```
$ aws ec2 delete-subnet --subnet-id subnet-021345abcdef6789 --region region-id
```

f. Descreva o gateway da Internet.

```
$ 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. Separe o gateway da Internet.

```
$ aws ec2 detach-internet-gateway \  
  --internet-gateway-id igw-1234567890abcdef \  
  --vpc-id vpc-021345abcdef6789 \  
  --region region-id
```

- h. Exclua o gateway da internet.

```
$ aws ec2 delete-internet-gateway \  
  --internet-gateway-id igw-1234567890abcdef \  
  --region region-id
```

- i. Exclua a VPC.

```
$ aws ec2 delete-vpc \  
  --vpc-id vpc-021345abcdef6789 \  
  --region region-id
```

- j. Exclua o segredo que contém a senha ReadOnlyUser.

```
$ aws secretsmanager delete-secret \  
  --secret-id arn:aws:secretsmanager:region-  
id:123456789012:secret:ADSecretPassword-1234 \  
  --region region-id
```

Configurando a criptografia de armazenamento compartilhado com uma chave AWS KMS

Saiba como configurar uma chave AWS KMS gerenciada pelo cliente para criptografar e proteger seus dados nos sistemas de armazenamento de arquivos em cluster para os AWS ParallelCluster quais estão configurados.

Ao usar a interface de linha de AWS ParallelCluster comando (CLI) ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

AWS ParallelCluster suporta as seguintes opções de configuração de armazenamento compartilhado:

- [SharedStorage](#) / [EbsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [EfsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [FsxLustreSettings](#) / [KmsKeyId](#)

Você pode usar essas opções para fornecer uma chave AWS KMS gerenciada pelo cliente para a criptografia do sistema de armazenamento compartilhado Amazon EBS, Amazon EFS e FSx for Lustre. Para usá-los, você deve criar e configurar uma política do IAM para o seguinte:

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

Pré-requisitos

- AWS ParallelCluster [está instalado](#).
- O AWS CLI [está instalado e configurado](#).
- Você tem um [par de chaves EC2](#).
- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.

Tópicos

- [Crie a política do](#)
- [Configurar e criar o cluster](#)

Crie a política do

Crie uma política.

1. Acesse o console do IAM: <https://console.aws.amazon.com/iam/home>
2. Escolha Políticas (Políticas).
3. Escolha Create policy (Criar política).
4. Selecione a guia JSON e cole a política a seguir. Certifique-se de substituir todas as ocorrências de **123456789012** pelo seu ID Conta da AWS e pela chave do nome do recurso da Amazon (ARN) Região da AWS e pelo seu próprio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:region-id:123456789012:key/abcd1234-ef56-gh78-ij90-
        abcd1234efgh5678"
      ]
    }
  ]
}
```

5. Nesse tutorial, crie um nome para a política `ParallelClusterKmsPolicy` e escolha Create Policy (Criar política).
6. Anote o ARN da política. Você precisará dele para configurar seu cluster.

Configurar e criar o cluster

Veja a seguir um exemplo de configuração de cluster que inclui um sistema de arquivos compartilhado Amazon Elastic Block Store com criptografia.

```
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/ebs1
    Name: shared-ebs1
    StorageType: Ebs
    EbsSettings:
      Encrypted: True
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

Substitua os itens em vermelho por seus próprios valores. Em seguida, crie um cluster que use sua chave AWS KMS para criptografar seus dados no Amazon EBS.

A configuração é semelhante para os sistemas de arquivos do Amazon EFS e FSx for Lustre.

A configuração SharedStorage do Amazon EFS é a seguinte.

```
...
SharedStorage:
  - MountDir: /shared/efs1
    Name: shared-efs1
    StorageType: Efs
    EfsSettings:
      Encrypted: True
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

A configuração SharedStorage do FSx for Lustre é a seguinte.

```
...
SharedStorage:
  - MountDir: /shared/fsx1
    Name: shared-fsx1
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      DeploymentType: PERSISTENT_1
      PerUnitStorageThroughput: 200
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

Executando trabalhos em um cluster de modo de várias filas

Este tutorial aborda como executar seu primeiro trabalho "Hello World" no AWS ParallelCluster com o [modo de múltiplas filas](#).

Ao usar a interface de linha de comando (CLI) AWS ParallelCluster ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria Free Tier AWS na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Pré-requisitos

- AWS ParallelCluster [está instalado](#).
- O AWS CLI [está instalado e configurado](#).
- Você tem um [par de chaves EC2](#).

- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.

Configure seu cluster

Primeiro, verifique se o AWS ParallelCluster está instalado corretamente, executando o comando a seguir.

```
$ pcluster version
```

Para ter mais informações sobre o `pcluster version`, consulte [pcluster version](#).

Esse comando retorna a versão do AWS ParallelCluster em execução.

Em seguida, execute `pcluster configure` para gerar um arquivo de configuração básico. Siga todas as instruções que seguem esse comando.

```
$ pcluster configure --config multi-queue-mode.yaml
```

Para obter mais informações sobre o comando `pcluster configure`, consulte [pcluster configure](#).

Depois de concluir esta etapa, um arquivo de configuração básico chamado `multi-queue-mode.yaml` é exibido. Esse arquivo contém uma configuração básica de cluster.

Na próxima etapa, você modifica seu novo arquivo de configuração e inicia um cluster com várias filas.

Note

Algumas instâncias que este tutorial usa não são elegíveis para o nível gratuito.

Para este tutorial, modifique seu arquivo de configuração para que corresponda à configuração a seguir. Os itens destacados em vermelho representam os valores do arquivo de configuração. Mantenha seus próprios valores.

```
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

```

Crie seu cluster do

Crie um cluster chamado `multi-queue-cluster` com base no seu arquivo de configuração.

```

$ 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": "3.7.0",
"clusterStatus": "CREATE_IN_PROGRESS"
}
}
```

Para obter mais informações sobre o comando `pcluster create-cluster`, consulte [pcluster create-cluster](#).

Para verificar o status do clustere, execute o comando a seguir.

```
$ 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": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

Quando o cluster é criado, o campo `clusterStatus` exibe `CREATE_COMPLETE`.

Faça o login no nó principal.

Use seu arquivo de chave SSH privada para fazer login no nó principal.

```
$ pcluster ssh --cluster-name multi-queue-cluster -i ~/path/to/yourkeyfile.pem
```

Para ter mais informações sobre o `pcluster ssh`, consulte [pcluster ssh](#).

Depois de fazer login, execute o comando `sinfo` para verificar se suas filas do programador estão definidas e configuradas.

Para obter mais informações sobre `sinfo`, consulte [sinfo](#) na documentação 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]
```

A saída mostra que você tem um `t2.micro` e um nó de computação `c5.xlarge` no estado `idle` que estão disponíveis no seu cluster.

Todos os outros nós estão no estado de economia de energia, indicado pelo sufixo `~` no estado do nó, sem nenhuma instância do EC2 suportando-os. A fila padrão é indicada por um sufixo `*` após o nome da fila. A `spot` é sua fila de trabalhos padrão.

Executar trabalho no modo de várias filas

Em seguida, tente fazer com que o trabalho fique em latência por um tempo. Posteriormente, o trabalho gera seu próprio nome de host. Certifique-se de que esse script possa ser executado pelo usuário atual.

```
$ 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
```

Envie o trabalho usando o comando `sbatch`. Solicite dois nós para esse trabalho com a opção `-N 2` e verifique se o trabalho foi enviado com êxito. Para obter mais informações sobre `sbatch`, consulte [sbatch](#) na documentação do Slurm.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 1
```

Agora, você pode visualizar a fila e verificar o status do trabalho com o comando `squeue`. Como você não especificou uma fila específica, a fila padrão (`spot`) é usada. Para obter mais informações sobre `squeue`, consulte [squeue](#) na documentação do Slurm.

```
$ 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
```

A saída mostra que, no momento, a tarefa está em um estado de execução. Aguarde o término do trabalho. Isso leva cerca de 30 segundos. Em seguida, execute `squeue` novamente.

```
$ squeue
JOBID PARTITION      NAME      USER      ST      TIME  NODES NODELIST(REASON)
```

Agora que todos os trabalhos na fila foram concluídos, procure o arquivo de saída chamado `slurm-1.out` em seu diretório atual.

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```

A saída mostra que o trabalho foi executado com êxito nos nós `spot-st-t2micro-1` e `spot-st-c5xlarge-1`.

Agora, envie o mesmo trabalho especificando restrições para instâncias específicas com os comandos a seguir.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 2
```

Você usou esses parâmetros para `sbatch`:

- `-N 3`— solicita três nós.
- `-p spot`— envia o trabalho para a fila `spot`. Você também pode enviar um trabalho para a fila `ondemand`, especificando `-p ondemand`.
- `-C "[c5.xlarge*1&t2.micro*2]"`— especifica as restrições específicas do nó para esse trabalho. Isso solicita que um nó `c5.xlarge` e dois nós `t2.micro` sejam usados para esse trabalho.

Execute o comando `sinfo` para visualizar os nós e as filas. As filas em AWS ParallelCluster são chamadas de partições no 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]

```

Os nós estão sendo ativados. Isso é indicado pelo sufixo # no estado do nó. Execute o comando `squeue` para visualizar informações sobre os trabalhos no 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

```

Seu trabalho está no estado CF (CONFIGURING), aguardando que as instâncias aumentem e se juntem ao cluster.

Após cerca de três minutos, os nós ficam disponíveis e o trabalho entra no estado 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

```

O trabalho termina e todos os três nós estão no estado `idle`.

```

$ 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]

```

Então, depois que nenhum trabalho permanecer na fila, verifique `slurm-2.out` no seu diretório local.

```

$ 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
```

Esse é o estado final do 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]
```

Depois de se desconectar do cluster, você pode limpá-lo, executando o `pcluster delete-cluster`. Para obter mais informações, consulte [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"
  }
}
```

Uso da API AWS ParallelCluster

Neste tutorial, você cria e testa a API com o [Amazon API Gateway](#) e um modelo do CloudFormation AWS ParallelCluster. Em seguida, você usa o cliente de exemplo disponível no GitHub para usar a API. Para obter mais informações sobre como usar a API, consulte a [API AWS ParallelCluster](#).

Este tutorial foi extraído do [workshop sobre HPC para clientes do setor público](#).

Ao usar a interface de linha de comando (CLI) AWS ParallelCluster ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria Free Tier AWS na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Pré-requisitos

- O AWS CLI é [instalado](#) e configurado no seu ambiente de computação.
- O AWS ParallelCluster é instalado em um ambiente virtual Para obter mais informações, consulte [Instalar AWS ParallelCluster em um ambiente virtual](#).
- Você tem um [par de chaves EC2](#).
- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.

Etapa 1: criar a API com o Amazon API Gateway

Fique no seu diretório de usuários domésticos e ative seu ambiente virtual:

1. Instale um processador de linha de comando JSON útil.

```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. Execute o comando a seguir para obter a versão AWS ParallelCluster e atribuí-la a uma variável de ambiente.

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" | tee -a ~/.bashrc
```


3. Crie uma variável de ambiente e atribua seu ID de região a ela.

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" |tee -a ~/.bashrc
```

4. Execute o comando a seguir para implantar a API.

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" |tee -a ~/.bashrc
```

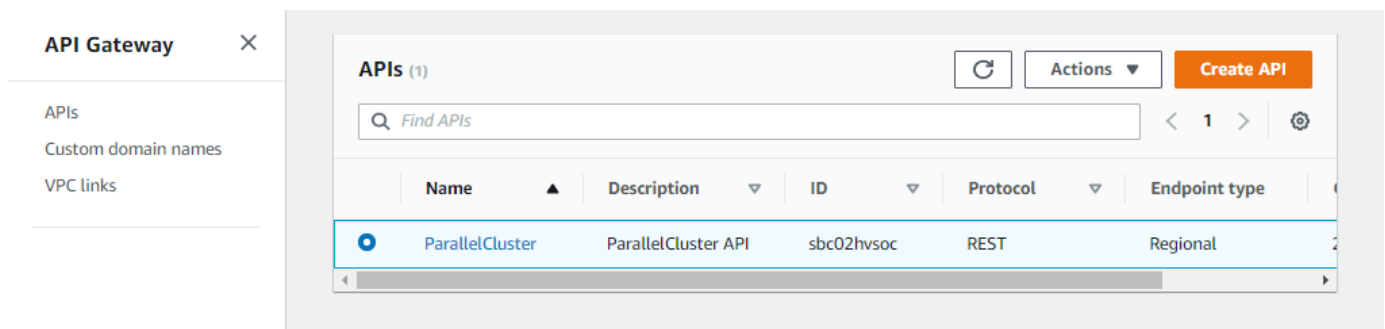
```
aws cloudformation create-stack \
  --region ${AWS_DEFAULT_REGION} \
  --stack-name ${API_STACK_NAME} \
  --template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
  ${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
  parallelcluster-api.yaml \
  --capabilities 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"
}
```

Depois que o processo estiver concluído, avance para a próxima etapa.

Etapa 2: testar a API no console do Amazon API Gateway

1. Faça login no AWS Management Console.
2. Vá até o [console do Amazon API Gateway](#).
3. Escolha sua implantação de API.



4. Escolha Stages e selecione um estágio.

The screenshot displays the AWS API Gateway console interface. The breadcrumb navigation at the top reads: Amazon API Gateway > APIs > ParallelCluster (sbc02hvsoc) > Stages > prod. The left-hand navigation pane lists various API Gateway components, with 'Stages' selected and highlighted. The main content area is titled 'prod Stage Editor' and features a 'Create' button. A prominent blue box at the top of the editor displays the 'Invoke URL: https://sbc02hvsoc.execute-api.us-east-1.amazonaws.com/prod'. Below this, there are several tabs for configuration: 'Settings' (selected), 'Logs/Tracing', 'Stage Variables', 'SDK Generation', and 'Export'. Under the 'Settings' tab, there are sub-sections for 'Cache Settings' (with an 'Enable API cache' checkbox), 'Default Method Throttling' (with an 'Enable throttling' checkbox and input fields for 'Rate' set to 100 and 'Burst' set to 10), 'Web Application Firewall (WAF)' (with a 'Web ACL' dropdown set to 'None'), and 'Client Certificate' (with a 'Certificate' dropdown set to 'None').

5. Observe a URL que o Gateway de API fornece para acessar ou invocar sua API. Está destacado em azul.
6. Escolha Recursos e selecione **GET em/clusters**.
7. Escolha o ícone de TESTE e, em seguida, role para baixo e escolha o ícone de TESTE.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET

Resources Actions **/v3/clusters - GET - Method Execution**

The screenshot displays the AWS API Gateway console for the endpoint `/v3/clusters`. On the left, a resource tree shows the hierarchy: `/v3` > `/clusters` > `GET`. The `GET` method is selected. In the center, a 'Client' box contains a 'TEST' button. To the right, the 'Method Request' box shows the following details: **Auth:** AWS IAM, **ARN:** `arn:aws:execute-api:us-east-1:123456789012:sbc02hvsoc/*/GET/v3/clusters`, and **Query Strings:** `region, nextToken, clusterStatus`. Below it, the 'Method Response' box contains the text 'Select an integration response.' Arrows indicate the flow from the Client to the Method Request and from the Method Response back to the Client.

A resposta à sua `/clusters` GET aparece.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > 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}

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

Etapa 3: preparar e testar um cliente de exemplo para invocar a API

Clone o código-fonte AWS ParallelCluster, o cd no diretório `api` e instale as bibliotecas de cliente do Python.

1.

```
$ 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
```

2. Navegue de volta ao seu diretório de usuário inicial.
3. Exporte a URL base do Gateway de API que o cliente usa durante a execução.

```
$ 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
```

4. Exporte um nome de cluster que o cliente usa para criar um cluster.

```
$ export CLUSTER_NAME="test-api-cluster"
echo "export CLUSTER_NAME=${CLUSTER_NAME}" |tee -a ~/.bashrc
```

5. Execute os comandos a seguir para armazenar as credenciais que o cliente de exemplo usa para acessar a 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
```

Etapa 4: copiar o script do código do cliente e executar testes de cluster

1. Copie o seguinte exemplo de código de cliente para o `test_pcluster_client.py` no seu diretório de usuário inicial. O código do cliente faz solicitações para fazer o seguinte:
 - Criar um cluster do
 - Descrever o cluster.
 - Listar os clusters.
 - Descrever a frota de computação.
 - Descrever as instâncias do 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. Criar uma configuração do cluster.

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

3. A biblioteca de cliente da API detecta automaticamente os detalhes da configuração de suas variáveis de ambiente (por exemplo `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, ou `AWS_SESSION_TOKEN`) ou `$HOME/.aws`. O comando a seguir muda seu perfil atual do IAM para o `ParallelClusterApiUserRole` designado.

```
$ 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"')
```

Erro a ser observado:

Se você vir um erro semelhante ao seguinte, você já presumiu que o `ParallelClusterApiUserRole` e o seu `AWS_SESSION_TOKEN` expiraram.

```

An error occurred (AccessDenied) when calling the AssumeRole operation:
User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/ApiTestSession
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::XXXXXXXXXXXX:role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

```

Elimine a função e execute novamente o comando `aws sts assume-role` para usar o `ParallelClusterApiUserRole`.

```
$ unset AWS_SESSION_TOKEN
```



```
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

Para fornecer suas permissões de usuário atuais para acesso à API, você deve [expandir a Política de Recursos](#).

4. Execute o seguinte comando para testar o cliente de exemplo.

```
$ 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!

```

Etapa 5: copiar o script do código do cliente e excluir cluster

1. Copie o seguinte código de cliente de exemplo para `delete_cluster_client.py`. O código do cliente faz uma solicitação para excluir o 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. Execute o comando a seguir para excluir o 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. Depois de terminar o teste, desfaça a definição das variáveis de ambiente.

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

Etapa 6: Limpar

Você pode usar o AWS Management Console ou a AWS CLI para excluir sua API.

1. No console AWS CloudFormation, escolha a pilha de API e, em seguida, escolha Excluir.
2. Se estiver usando o AWS CLI, execute o comando a seguir.

Usar o AWS CloudFormation.

```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```

Criação de um cluster com contabilidade Slurm

Saiba como configurar e criar um cluster com contabilidade Slurm. Para ter mais informações, consulte [Slurmcontabilidade com AWS ParallelCluster](#).

Ao usar a interface de linha de AWS ParallelCluster comando (CLI) ou a API, você paga apenas pelos AWS recursos criados ao criar ou atualizar AWS ParallelCluster imagens e clusters. Para ter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A AWS ParallelCluster interface do usuário é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para ter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Neste tutorial, você usa um [modelo de criação CloudFormation rápida \(us-east-1\) para criar um banco de dados sem servidor para MySQL](#). [Amazon Aurora](#) O modelo instrui CloudFormation a criar todos os componentes necessários para implantar um banco de dados Amazon Aurora sem servidor na mesma VPC do cluster. O modelo também cria uma configuração básica de rede e segurança para a conexão entre o cluster e o banco de dados.

Note

A partir da versão 3.3.0, AWS ParallelCluster oferece suporte à Slurm contabilização com o parâmetro de configuração de cluster [SlurmSettings/Database](#).

Note

O modelo de criação rápida serve como exemplo. Esse modelo não abrange todos os casos de uso possíveis de um servidor de banco de dados de contabilidade Slurm. É sua responsabilidade criar um servidor de banco de dados com a configuração e a capacidade adequadas para suas workloads de produção.

Pré-requisitos:

- AWS ParallelCluster [está instalado](#).
- O AWS CLI [está instalado e configurado](#).
- Você tem um [par de chaves EC2](#).
- Você tem um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.
- A região na qual você implanta o modelo de criação rápida oferece suporte à Amazon Aurora MySQL serverless v2. Para ter mais informações, consulte [Aurora Serverless v2 com Aurora MySQL](#).

Etapa 1: criar a VPC e as sub-redes para AWS ParallelCluster

Para usar o CloudFormation modelo fornecido para o banco de dados Slurm contábil, você deve ter a VPC do cluster pronta. Você pode fazer isso manualmente ou como parte do procedimento [Configure e crie um cluster com a interface de linha de AWS ParallelCluster comando](#). Se você já usou o AWS ParallelCluster, talvez tenha uma VPC pronta para a implantação do cluster e do servidor de banco de dados.

Etapa 2: criar a pilha de banco de dados

Use o [modelo de criação CloudFormation rápida \(us-east-1\) para criar uma pilha de banco de dados](#) para contabilidade. Slurm O modelo requer as seguintes entradas:

- Credenciais do servidor de banco de dados, especificamente o nome de usuário e senha do administrador.
- Dimensionamento do cluster sem Amazon Aurora servidor. Isso depende do carregamento esperado do cluster.
- Parâmetros de rede, especificamente a VPC e as sub-redes de destino ou blocos CIDR para a criação das sub-redes.

Selecione as credenciais e o tamanho apropriados para seu servidor de banco de dados. Para as opções de rede, você precisa usar a mesma VPC na qual o cluster AWS ParallelCluster está implantado. Você pode criar as sub-redes para o banco de dados e passá-las como entrada para o modelo. Ou forneça dois blocos CIDR separados para as duas sub-redes e deixe o CloudFormation modelo criar as duas sub-redes para blocos CIDR. Certifique-se de que os blocos CIDR não

se sobreponham às sub-redes existentes. Se os blocos CIDR se sobrepuserem às sub-redes existentes, a pilha não será criada.

O servidor de banco de dados leva vários minutos para ser criado.

Etapa 3: criar um cluster com a contabilidade Slurm ativada

O CloudFormation modelo fornecido gera uma CloudFormation pilha com algumas saídas definidas. A partir do AWS Management Console, você pode visualizar as saídas na guia Saídas na visualização da CloudFormation pilha. Para habilitar a contabilidade Slurm, algumas dessas saídas devem ser usadas no arquivo de configuração do cluster do AWS ParallelCluster :

- DatabaseHost: usado para o parâmetro de configuração do cluster [SlurmSettings / Database / Uri](#).
- DatabaseAdminUser: usado para o valor do parâmetro de configuração do cluster [SlurmSettings / Database / UserName](#).
- DatabaseSecretArn: usado para o parâmetro de configuração do cluster [SlurmSettings / Database / PasswordSecretArn](#).
- DatabaseClientSecurityGroup: esse é o grupo de segurança anexado ao nó do cabeçalho do cluster definido no parâmetro de configuração [HeadNode / Networking / SecurityGroups](#).

Atualize os parâmetros Database do arquivo de configuração do cluster com os valores de saída. Use a [pcluster](#) CLI para criar um cluster.

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

Depois que o cluster for criado, você poderá começar a usar comandos de contabilidade Slurm, como `sacctmgr` ou `sacct`.

Revertendo para uma versão anterior do documento do AWS Systems Manager

Saiba como reverter para uma versão anterior do documento do AWS Systems Manager. Para obter mais informações, consulte [Documentos do AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager.

Ao usar a interface de linha de AWS ParallelCluster comando (CLI) ou a API, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

A interface do usuário AWS ParallelCluster é construída em uma arquitetura sem servidor e você pode usá-la na categoria AWS Free Tier na maioria dos casos. Para obter mais informações, consulte [Custos de interface de usuário do AWS ParallelCluster](#).

Pré-requisitos:

- Um Conta da AWS com permissões para gerenciar documentos SSM.
- O AWS CLI [está instalado e configurado](#).

Reverter para uma versão anterior do documento SSM

1. No seu terminal, execute o comando a seguir para obter a lista de documentos SSM existentes que você possui.

```
$ aws ssm list-documents --document-filter "key=Owner,value=Self"
```

2. Reverter para uma versão anterior do documento SSM. Neste exemplo, revertemos para uma versão anterior do documento `SessionManagerRunShell`. Você pode usar o documento `SessionManagerRunShell` SSM para personalizar cada sessão de shell SSM que você inicia.
 - a. Encontre o parâmetro `DocumentVersion` para `SessionManagerRunShell` executando o seguinte 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": []
}
}
```

A versão mais recente é 2.

- b. Reverta para a versão anterior executando o seguinte comando:

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-  
version 2
```

3. Verifique se a versão do documento foi revertida executando o comando `describe-document` novamente:

```
$ 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": []
}
}
```

A versão mais recente é 1.

Criação de um cluster com AWS CloudFormation

Saiba como criar um cluster com um recurso AWS ParallelCluster CloudFormation personalizado. Para ter mais informações, consulte [AWS CloudFormation recurso personalizado](#).

Ao usar AWS ParallelCluster, você paga apenas pelos AWS recursos criados ao criar ou atualizar AWS ParallelCluster imagens e clusters. Para ter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

Pré-requisitos:

- O AWS CLI [está instalado e configurado](#).
- Um [par de chaves do EC2](#).
- Um perfil do IAM com as [permissões](#) necessárias para executar a [pcluster](#) CLI.

Criação de clusters com uma pilha de criação CloudFormation rápida

Neste tutorial, você usa uma pilha de criação rápida para implantar um CloudFormation modelo que cria um cluster e os seguintes recursos: AWS

- Uma CloudFormation pilha raiz criada usando uma pilha de criação CloudFormation rápida.
- CloudFormation Pilhas aninhadas que incluem políticas padrão, configuração padrão de VPC e um provedor de recursos personalizado.
- Um exemplo AWS ParallelCluster de pilha de clusters e um cluster no qual você pode fazer login e executar trabalhos.

Crie um cluster com AWS CloudFormation

1. Faça login no AWS Management Console.
2. Abra o [link CloudFormation de criação rápida](#) para criar os seguintes recursos no CloudFormation console:
 - Uma CloudFormation pilha aninhada com uma VPC com uma sub-rede pública e privada para executar o nó principal do cluster e os nós de computação, respectivamente.
 - Uma CloudFormation pilha aninhada com um recurso AWS ParallelCluster personalizado para gerenciar o cluster.
 - Uma CloudFormation pilha aninhada com as políticas padrão para gerenciar o cluster.
 - Uma CloudFormation pilha raiz para as pilhas aninhadas.
 - Um AWS ParallelCluster cluster com o Slurm agendador e um número definido de nós de computação.

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
cluster-0

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
us-east-2a

KeyName
KeyPair to login to the head node
Select AWS::EC2::KeyPair::KeyName

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. Na seção Parâmetros de pilha de criação rápida, insira valores para os seguintes parâmetros:
 - a. Para KeyName, insira o nome do seu key pair do EC2.
 - b. Para AvailabilityZone, escolha uma AZ para seus nós de cluster, por exemplo, us-east-1a.

4. Na parte inferior da página, marque as caixas de seleção que reconhecem cada um dos recursos de acesso.
5. Selecione Criar pilha.
6. Aguarde até que a CloudFormation pilha alcance o CREATE_COMPLETE estado.

Criação de clusters com a interface de linha de AWS CloudFormation comando (CLI)

Neste tutorial, você usa a Interface de Linha de AWS Comando (CLI) para CloudFormation implantar um CloudFormation modelo que cria um cluster.

Crie os seguintes AWS recursos:

- Uma CloudFormation pilha raiz criada usando uma pilha de criação CloudFormation rápida.
- CloudFormation Pilhas aninhadas que incluem políticas padrão, configuração padrão de VPC e um provedor de recursos personalizado.
- Um exemplo AWS ParallelCluster de pilha de clusters e um cluster no qual você pode fazer login e executar trabalhos.

Substitua *as entradas destacadas em vermelho*, como *ketpair*, por seus próprios valores.

Crie um cluster com AWS CloudFormation

1. Crie um CloudFormation modelo chamado `cluster_template.yaml` com o seguinte conteúdo:

```
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: 3.7.0

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 ]

```

- Execute o seguinte comando da AWS CLI para implantar a CloudFormation pilha para criação e gerenciamento de clusters.

```

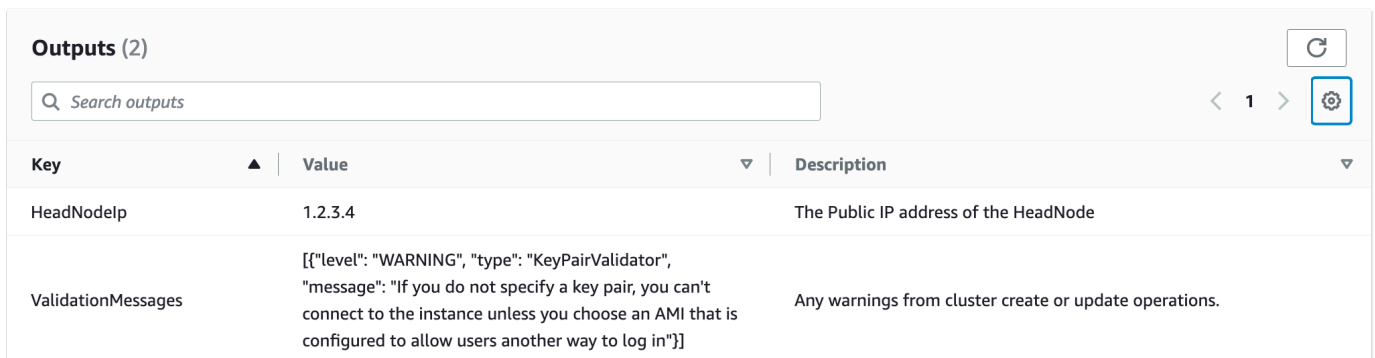
$ aws cloudformation deploy --template-file ./cluster_template.yaml \
  --stack-name mycluster \
  --parameter-overrides KeyName=keypair \
    AvailabilityZone=us-east-2b \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND

```

Exibir saída CloudFormation do cluster

Visualize a saída do CloudFormation cluster para obter detalhes úteis do cluster. A propriedade `ValidationMessages` adicionada fornece acesso às mensagens de validação das operações de criação e atualização do cluster.

- Navegue até o [CloudFormation console](#) e selecione a pilha que inclui seu recurso AWS ParallelCluster personalizado.
- Escolha Stack details (Detalhes da pilha) e selecione a guia Outputs (Saídas).



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.

As mensagens de validação poderão estar truncadas. Para obter mais informações sobre recuperação de logs, consulte [Solução de problemas do AWS ParallelCluster](#).

Acessar seu cluster

Acessar o cluster.

ssh no nó principal do cluster

1. Depois que a implantação da CloudFormation pilha for concluída, obtenha o endereço IP do nó principal com o seguinte comando:

```
$ HEAD_NODE_IP=$(aws cloudformation describe-stacks --stack-name=mycluster --query "Stacks|[0].Outputs[?OutputKey=='HeadNodeIp']|[0].OutputValue" --output=text)
```

Você também pode recuperar o endereço IP do nó principal a partir do HeadNodeIpparâmetro na guia Saídas da pilha de clusters no console. CloudFormation

Você pode encontrar o endereço IP do nó principal aqui porque ele foi adicionado na Outputs seção do CloudFormation modelo de cluster, especificamente para este exemplo de cluster.

2. Conecte-se ao nó principal do cluster executando o comando a seguir:

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

Limpeza

Excluir o cluster.

1. Execute o seguinte comando da AWS CLI para excluir a CloudFormation pilha e o cluster.

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. Verifique o status de exclusão da pilha executando o seguinte comando.

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

AWS ParallelClusterIntegração da interface do usuário com o Identity Center

O objetivo deste tutorial é demonstrar como integrar a AWS ParallelCluster interface do usuário com o IAM Identity Center para uma solução de login único que unifica os usuários no Active Directory e pode ser compartilhada com clusters. AWS ParallelCluster

Ao usar o AWS ParallelCluster, você paga apenas pelos recursos AWS criados ao criar ou atualizar imagens e clusters AWS ParallelCluster. Para obter mais informações, consulte [Serviços AWS usados por AWS ParallelCluster](#).

Pré-requisitos:

- Uma AWS ParallelCluster interface de usuário existente que pode ser instalada seguindo as instruções [aqui](#).
- Um Active Directory gerenciado existente, de preferência um que você também usará para [integração com AWS ParallelCluster](#).

Habilitar o IAM Identity Center

Se você já tem um centro de identidade conectado ao seu AWS Managed Microsoft AD (Active Directory), ele pode ser usado e você pode pular para a seção Adicionar seu aplicativo ao IAM Identity Center.

Se você ainda não tiver uma central de identidade conectada a umaAWS Managed Microsoft AD, siga as etapas abaixo para configurá-la.

Habilitando o Identity Center

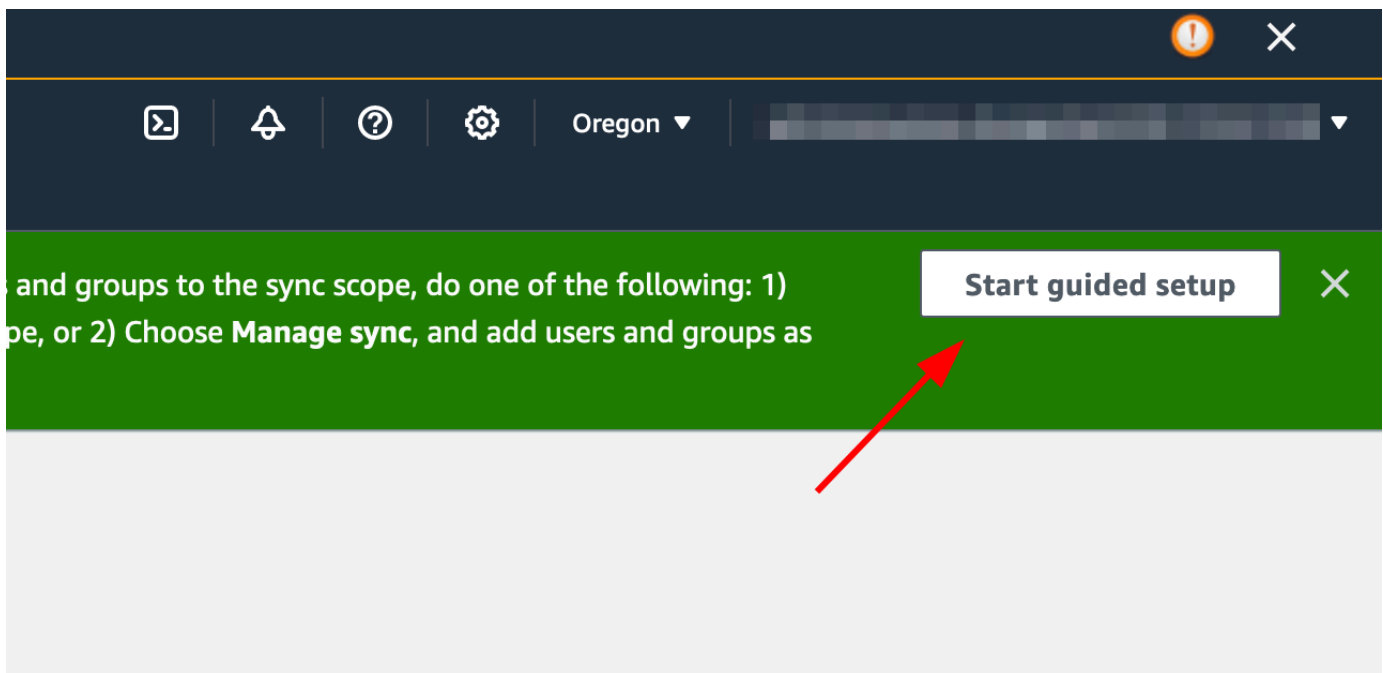
1. No console, navegue até o IAM Identity Center. (Verifique se você está na região em que você tem o seuAWS Managed Microsoft AD.)
2. Clique no botão Habilitar, isso pode perguntar se você deseja habilitar organizações. Esse é um requisito para que você possa optar por habilitá-lo. Observação: Isso enviará um e-mail ao administrador da sua conta com um e-mail de confirmação que você deve seguir o link para confirmar.

Conectando o Identity Center ao AD gerenciado

1. Na próxima página, depois de ativar a central de identidade, você deve ver as Etapas de configuração recomendadas. Na Etapa 1, selecione Escolha sua fonte de identidade.
2. Na seção Fonte de identidade, clique no menu suspenso Ações (no canto superior direito) e selecione Alterar fonte de identidade.
3. Selecione Active Directory.
4. Em Diretórios existentes, escolha seu diretório.
5. Clique em Next.
6. Revise suas alterações, vá até o final, digite ACEITAR na caixa de texto para confirmar e clique em Alterar fonte de identidade.
7. Aguarde a conclusão das alterações e, em seguida, você verá um banner verde na parte superior.

Sincronizando usuários e grupos com o Identity Center

1. No banner verde, clique em Iniciar configuração guiada (botão no canto superior direito)



2. Em Configurar mapeamentos de atributos, clique em Avançar
3. Na seção Configurar escopo de sincronização, digite o nome dos usuários que você deseja sincronizar com a central de identidade e clique em Adicionar
4. Quando terminar de adicionar usuários e grupos, clique em Avançar

Users | **Groups**

User

corp.pcluster.com ▼ 🔍 Enter exact match user to add to sync scope **Add**

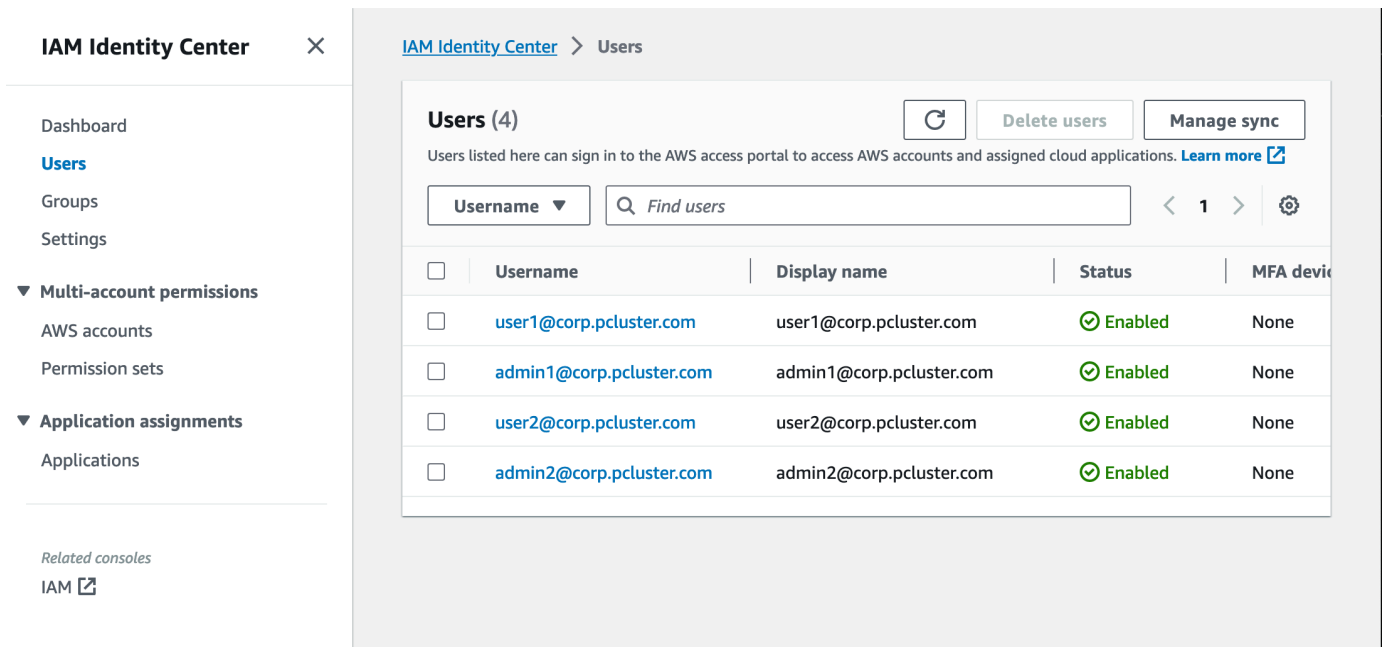
Added users and groups (4) **Remove**

<input type="checkbox"/>	Username / Group name	Type	Domain
<input type="checkbox"/>	user1	User	corp.pcluster.com
<input type="checkbox"/>	user2	User	corp.pcluster.com
<input type="checkbox"/>	admin1	User	corp.pcluster.com
<input type="checkbox"/>	admin2	User	corp.pcluster.com

Cancel **Previous** **Next**

5. Revise suas alterações e clique em Salvar configuração
6. Se você ver um aviso na próxima tela sobre os usuários não estarem sincronizados, selecione o botão Retomar sincronização no canto superior direito.
7. Em seguida, para habilitar usuários, na guia Usuários à esquerda, selecione um usuário e clique em Habilitar acesso do usuário > Habilitar acesso do usuário

Observação: talvez seja necessário selecionar Retomar sincronização se você tiver um banner de aviso na parte superior e esperar que os usuários sincronizem (experimente o botão Atualizar para ver se eles já estão sincronizados).



The screenshot shows the IAM Identity Center console interface. On the left is a navigation menu with options like Dashboard, Users, Groups, Settings, Multi-account permissions, and Application assignments. The main content area is titled 'IAM Identity Center > Users' and displays a list of 4 users. At the top of the list are buttons for 'Delete users' and 'Manage sync'. Below the buttons is a search bar labeled 'Find users' and a dropdown menu for 'Username'. The table below has columns for Username, Display name, Status, and MFA device. All users are listed with a status of 'Enabled' and 'None' for MFA device.

<input type="checkbox"/>	Username	Display name	Status	MFA device
<input type="checkbox"/>	user1@corp.pcluster.com	user1@corp.pcluster.com	Enabled	None
<input type="checkbox"/>	admin1@corp.pcluster.com	admin1@corp.pcluster.com	Enabled	None
<input type="checkbox"/>	user2@corp.pcluster.com	user2@corp.pcluster.com	Enabled	None
<input type="checkbox"/>	admin2@corp.pcluster.com	admin2@corp.pcluster.com	Enabled	None

Adicionar seu aplicativo ao IAM Identity Center

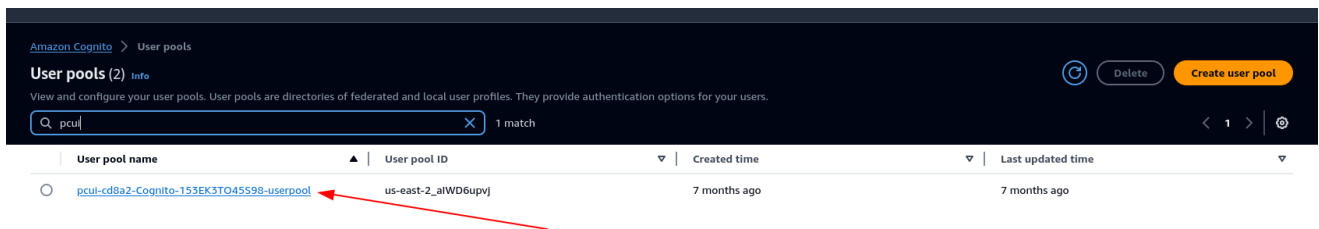
Depois de sincronizar seus usuários com o IAM Identity Center, você precisará adicionar um novo aplicativo. Isso configura quais aplicativos habilitados para SSO estarão disponíveis no portal do IAM Identity Center. Nesse caso, adicionaremos a AWS ParallelCluster interface do usuário como um aplicativo, enquanto o IAM Identity Center será o provedor de identidade.

A próxima etapa adicionará a AWS ParallelCluster interface do usuário como um aplicativo no IAM Identity Center. AWS ParallelCluster A interface do usuário é um portal da web que ajuda o usuário a gerenciar seus clusters. Para obter mais informações, consulte [AWS ParallelClusterUI](#).

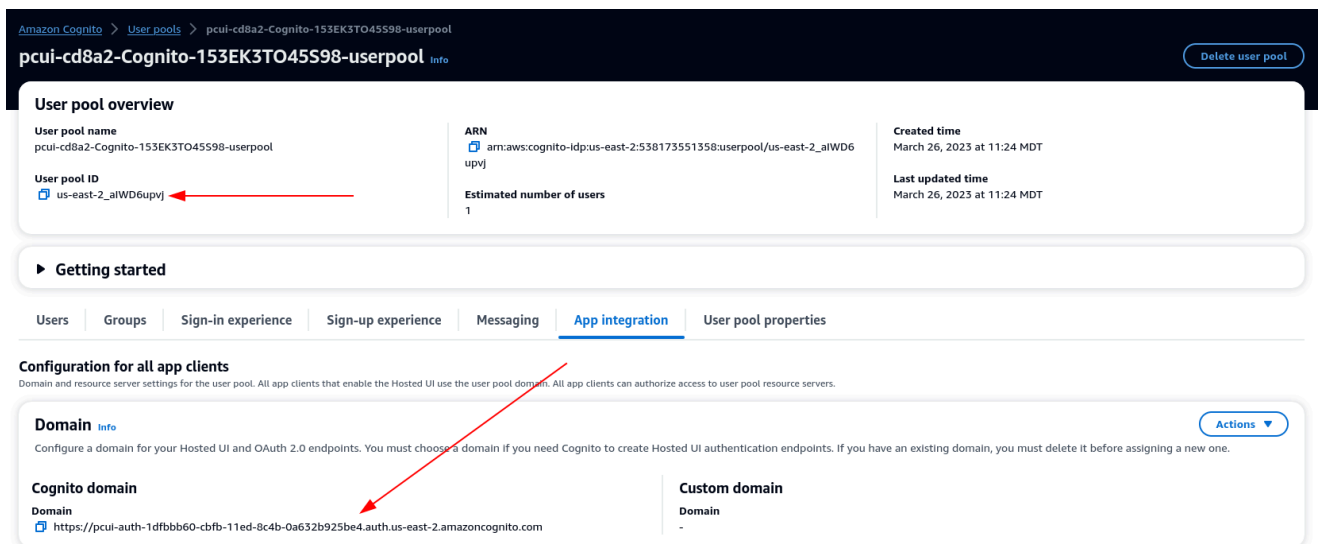
Configurando o aplicativo no Identity Center

1. Em IAM Identity Center > Aplicativos (encontrado na barra de menu à esquerda, clique em Aplicativos)
2. Clique em Adicionar aplicativo
3. Selecione Adicionar aplicativo SAML 2.0 personalizado
4. Clique em Avançar
5. Selecione o nome de exibição e a descrição que você gostaria de usar (por exemplo, PCUI e AWS ParallelCluster UI)
6. Em Metadados do IAM Identity Center, copie o link para o arquivo de metadados SAML do IAM Identity Center e salve para mais tarde. Isso será usado ao configurar o SSO no aplicativo web

7. Em Propriedades do aplicativo, na URL inicial do aplicativo, coloque seu endereço PCUI. Isso pode ser encontrado acessando o CloudFormation console, selecionando a pilha que corresponde à PCUI (por exemplo, parallelcluster-ui) e indo até a guia Saídas para encontrar UiURL ParallelCluster
por exemplo, <https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com>
8. Em Metadados da aplicação, escolha Digite manualmente seus valores de metadados. Em seguida, forneça os seguintes valores.
 - a. Importante: certifique-se de substituir os valores domain-prefix, region e userpool-id por informações específicas do seu ambiente.
 - b. O prefixo do domínio, a região e o userpool-id podem ser obtidos abrindo o console Amazon Cognito > Grupos de usuários



- c. Selecione o grupo de usuários que corresponde ao PCUI (que terá um nome de grupo de usuários como PCUI-cd8a2-cognito-153ek3to45s98-userpool)
- d. Navegue até Integração de aplicativos



9. <domain-prefix>URL do Application Assertion Consumer Service (ACS): <https://.auth.<region>.amazoncognito.com/saml2/idresponse>

Público SAML do aplicativo: urn:amazon:cognito:sp: <userpool-id>

10. Selecione Enviar. Em seguida, acesse a página Detalhes do aplicativo que você adicionou.
11. Selecione a lista suspensa Ações e escolha Editar mapeamentos de atributos. Em seguida, forneça os seguintes atributos.
 - a. Atributo do usuário no aplicativo: assunto (Nota: o assunto está pré-preenchido.) → Mapeia esse valor de string ou atributo de usuário no IAM Identity Center: `{user:email}`, Formato: EmailAddress
 - b. Atributo do usuário no aplicativo: email → Mapeia para esse valor de string ou atributo do usuário no IAM Identity Center: `{user:email}`, Formato: não especificado

Attribute mappings for PCUI

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. [Learn more](#)

User attribute in the application	Maps to this string value or user attribute in IAM Identity Center	Format	
Subject	<code>{user:email}</code>	emailAddress	
email	<code>{user:email}</code>	unspecified	Remove
Add new attribute mapping			

[Cancel](#) [Save changes](#)


12. Salve as alterações.

13. Escolha o botão Atribuir usuários e, em seguida, atribua seu usuário ao aplicativo. Esses são os usuários em seu Active Directory que terão acesso à interface PCUI.

[IAM Identity Center](#) > [Applications](#) > PCUI

PCUI

Details Actions ▾

	Display name PCUI	Description AWS ParallelCluster UI
---	----------------------	---------------------------------------

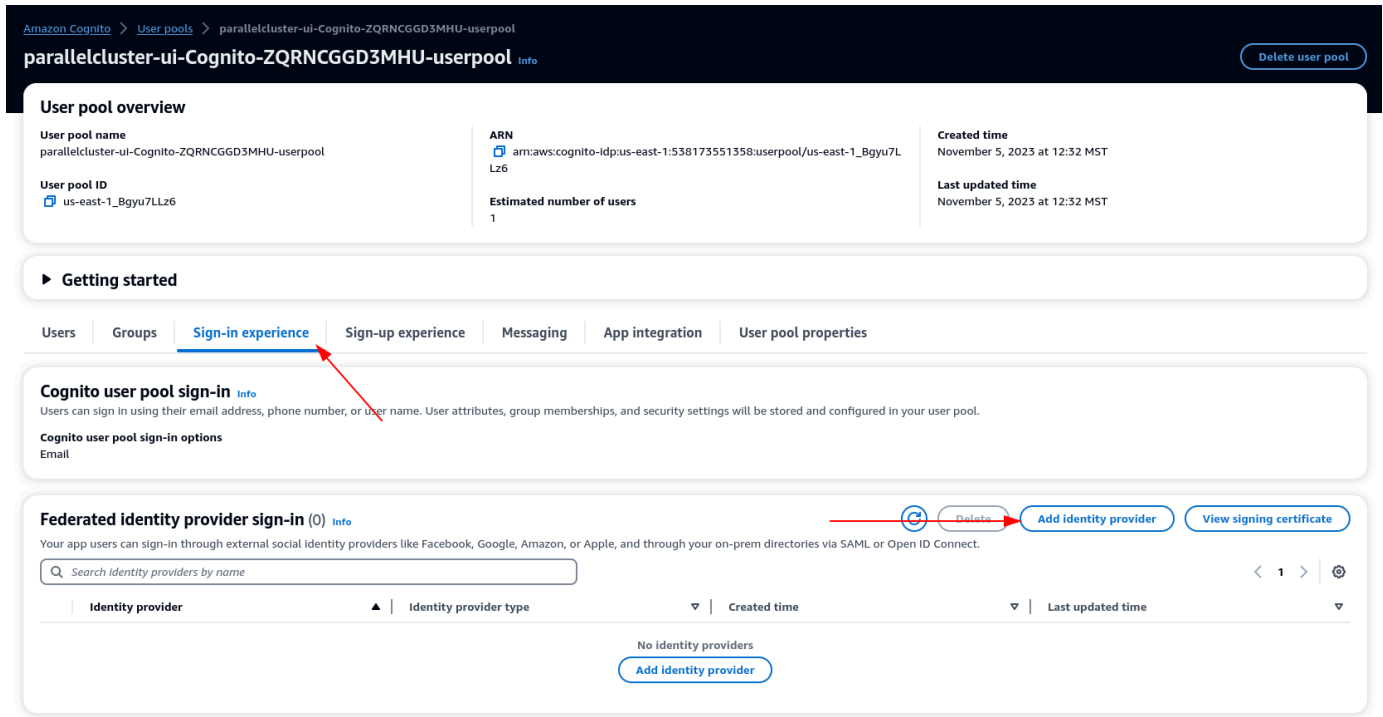
Assigned users (1) [Remove access](#) [Assign Users](#)

The following users and groups from your connected directory can access this application. [Learn more](#)

	User/Group name	Type
<input type="checkbox"/>	admin1@corp.pcluster.com	User

Configure o IAM Identity Center como um IdP SAML em seu grupo de usuários

1. Nas configurações do seu grupo de usuários, selecione Experiência de login > Adicionar provedor de identidade



The screenshot shows the AWS IAM console interface for a user pool. The breadcrumb navigation is **Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool**. The user pool name is **parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool** and the ID is **us-east-1_Bgyu7LLz6**. The ARN is **arn:aws:cognito-idp:us-east-1:538173551358:userpool/us-east-1_Bgyu7LLz6**. The estimated number of users is **1**. The user pool was created on **November 5, 2023 at 12:32 MST** and last updated on **November 5, 2023 at 12:32 MST**. The **Getting started** section includes tabs for **Users**, **Groups**, **Sign-in experience** (selected), **Sign-up experience**, **Messaging**, **App integration**, and **User pool properties**. The **Cognito user pool sign-in** section explains that users can sign in using email, phone number, or user name. The **Cognito user pool sign-in options** section shows **Email** as an option. The **Federated identity provider sign-in (0)** section allows adding external identity providers like Facebook, Google, Amazon, or Apple. It includes a search bar for identity providers by name and a table with columns for **Identity provider**, **Identity provider type**, **Created time**, and **Last updated time**. The table currently shows **No identity providers** and an **Add identity provider** button. A red arrow points to the **Add identity provider** button in the **Federated identity provider sign-in** section.

2. Escolha um IdP SAML
3. Para o nome do provedor, forneça IdentityCenter
4. Em Fonte do documento de metadados, escolha Inserir URL do endpoint do documento de metadados e forneça o URL copiado durante a configuração do aplicativo do Identity Center.
5. Em Atributos, para e-mail, escolha e-mail

SAML

Configure a SAML 2.0 identity provider for your user pool.

Register your app with your SAML provider

To connect a SAML provider to Cognito, add your user pool as a relying party or application with your SAML 2.0 identity provider, and upload a metadata document to Cognito.

Set up SAML federation with this user pool

Provider name [Info](#)
Enter a friendly name for your SAML 2.0 identity provider.

Identifiers - optional [Info](#)
Enter identifiers for this provider. Identifiers can be used to redirect users to the correct IdP in multitenant apps.

Separate each identifier by a comma

Sign-out flow [Info](#)
 Add sign-out flow
Enable simultaneous sign-out from the SAML provider and Cognito.

Metadata document source [Info](#)
Provide a SAML metadata document. This document is issued by your SAML provider. It includes the issuer's name, expiration information, and keys that can be used to validate the response from the identity provider.

Upload metadata document
 Enter metadata document endpoint URL

Enter metadata document endpoint URL [Info](#)

Map attributes between your SAML provider and your user pool

Your required attributes are mapped to the equivalent SAML attributes. Each attribute you add must be mapped to a SAML attribute.

User pool attribute	SAML attribute
email	email

[Add another attribute](#)

[Cancel](#) [Add identity provider](#)

6. Selecione Adicionar provedor de identidade.

Integre o IdP com o cliente do aplicativo do grupo de usuários

1. Em seguida, na seção Integração de aplicativos do seu grupo de usuários, escolha o cliente listado em Lista de clientes de aplicativos

Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool

parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool Info

[Delete user pool](#)

User pool overview

User pool name parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool	ARN arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_Bgyu7Lz6	Created time November 5, 2023 at 12:32 MST
User pool ID us-east-1_Bgyu7Lz6	Estimated number of users 1	Last updated time November 5, 2023 at 12:32 MST

Getting started

Users | Groups | Sign-in experience | Sign-up experience | Messaging | **App integration** | User pool properties

Configuration for all app clients

Domain and resource server settings for the user pool. All app clients that enable the Hosted UI use the user pool domain. All app clients can authorize access to user pool resource servers.

Domain Info

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. You must choose a domain if you need Cognito to create Hosted UI authentication endpoints. If you have an existing domain, you must delete it before assigning a new one.

[Actions](#)

Cognito domain	Custom domain
Domain https://pcul-auth-06f29200-7c12-11ee-b755-0e11297ecb0d.auth.us-east-1.amazoncognito.com	Domain -

Resource servers (0) Info

Configure resource servers. A resource server is a remote server that authorizes access based on OAuth 2.0 scopes in an access token.

Resource server name	Resource server identifier	Custom scopes
No resource servers		

[Create resource server](#)

App client defaults

Hosted UI customization and advanced security settings for the user pool. You can customize the Hosted UI and advanced security in app clients to override the defaults.

Hosted UI customization Info

Customize the hosted sign-up and sign-in pages to match your app's style and branding by uploading your own logo and customized CSS.

[Edit](#)

Logo -	Custom CSS -
------------------	------------------------

Advanced security Info

Configure advanced security features, including Cognito's automatic responses to suspicious user activity. Advanced security adds cost to your bill. [See pricing](#)

[Enable](#)

Status
 Disabled

App client list

The app clients that integrate your apps with your user pool. Configure client overrides to user pool default configurations, and configure Amazon Pinpoint analytics.

App clients and analytics (1) Info

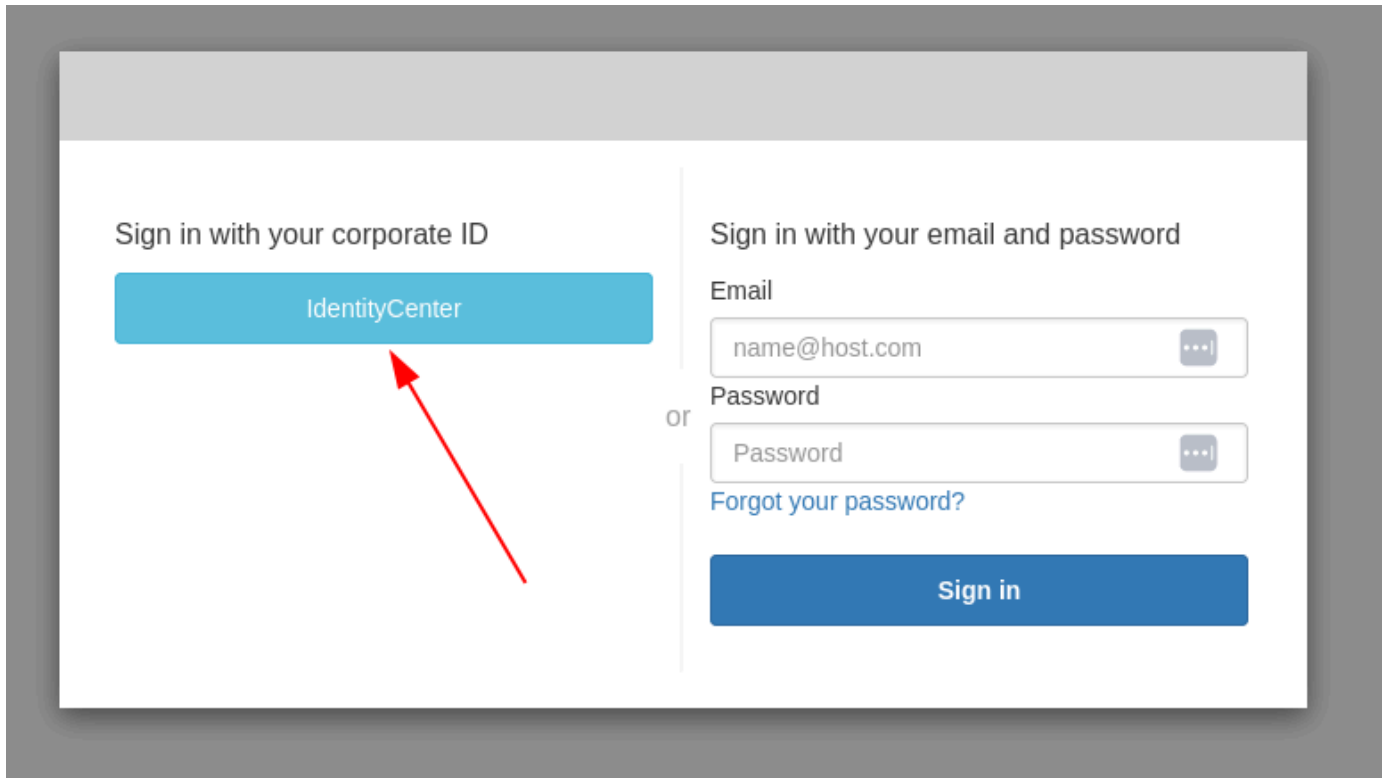
Configure an app client. App clients are the user pool authentication resources attached to your app. Select an app client to configure the permitted authentication actions for an app.

App client name	Client ID
<input type="radio"/> CognitoAppClient-ETqXbqL5wRVs	...

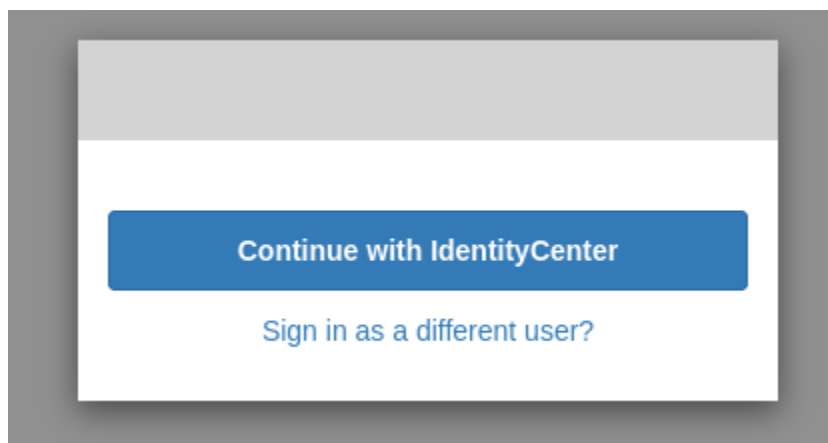
2. Em Interface do usuário hospedada, escolha Editar
3. Em Provedores de identidade IdentityCenter, escolha também.
4. Selecione Save changes (Salvar alterações)

Valide sua configuração

1. Em seguida, validaremos a configuração que acabamos de criar fazendo login no PCUI. Faça login no seu portal PCUI e agora você verá uma opção para entrar com sua ID corporativa:



2. Ao clicar no IdentityCenterbotão, você deve acessar o login do IdP do IAM Identity Center, seguido por uma página com seus aplicativos, que inclui o PCUI. Abra esse aplicativo.
3. Depois de acessar a tela a seguir, seu usuário será adicionado ao grupo de usuários do Cognito.



Torne seu usuário um administrador

1. Agora, navegue até o console Amazon Cognito > Grupos de usuários e selecione o usuário recém-criado, que deve ter o prefixo identitycenter.

parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool [Info](#) [Delete user pool](#)

User pool overview

User pool name parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool	ARN am:aws:cognito-idp:us-east-1:000000000000:userpool/us-east-1_Bgyu7Lz6	Created time November 5, 2023 at 12:32 MST
User pool ID us-east-1_Bgyu7LLz6	Estimated number of users 2	Last updated time November 5, 2023 at 12:32 MST

Getting started

[Users](#) [Groups](#) [Sign-in experience](#) [Sign-up experience](#) [Messaging](#) [App integration](#) [User pool properties](#)

Users (2) [Info](#)

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

Property:

User name	Email address	Email verified	Confirmation status	Status
c8f7eccc-e647-4227-afa2-080274ebfefe	user@amazon.com	Yes	Confirmed	Enabled
identitycenter_admin1@corp.pcluster.c...	admin1@corp.pcluster.com	No	External provider	Enabled

Import users (0) [Info](#)

View and create user CSV Import jobs. Amazon Cognito can import users into this user pool from a specially-formatted CSV file. You can't import user passwords.

Job name	Status	Imported users	Skipped users	Failed users	CloudWatch logs	Created time
----------	--------	----------------	---------------	--------------	-----------------	--------------

No user import jobs found

[Create import job](#)

2. Em Associações de grupo, selecione Adicionar usuário ao grupo, escolha administrador e clique em Adicionar.
3. Agora, ao clicar em Continuar com, IdentityCenter você será direcionado para a página da AWS ParallelCluster interface do usuário.

Solução de problemas do AWS ParallelCluster

A AWS ParallelCluster comunidade mantém uma página Wiki que fornece muitas dicas de solução de problemas na [AWS ParallelCluster GitHub Wiki](#). Para obter uma lista de problemas conhecidos, consulte [Problemas conhecidos](#).

Tópicos

- [Tentando criar um cluster](#)
- [Tentar executar um trabalho](#)
- [Tentando atualizar um cluster](#)
- [Tentar acessar o armazenamento](#)
- [Tentando excluir um cluster](#)
- [Tentando atualizar a pilha de APIs AWS ParallelCluster](#)
- [Vendo erros nas inicializações dos nós de computação](#)
- [Métricas de integridade do cluster para solução de problemas](#)
- [Solução de problemas de implantação de cluster](#)
- [Solucionar problemas de escala](#)
- [Grupos de posicionamento e problemas de execução de instâncias](#)
- [Diretórios que não podem ser substituídos](#)
- [Solucionar problemas no NICE DCV](#)
- [Solução de problemas em clusters com integração AWS Batch](#)
- [Solução de problemas de integração de vários usuários com o Active Directory](#)
- [Solução de problemas de AMI personalizada](#)
- [Solução de problemas de tempo limite de atualização de cluster quando cfn-hup não está em execução](#)
- [Solução de problemas de rede](#)
- [Falha na atualização do cluster na ação personalizada onNodeUpdated](#)
- [Vendo erros com a configuração personalizada Slurm](#)
- [Alarmes de cluster](#)
- [Suporte adicional](#)

Tentando criar um cluster

Ao usar a AWS ParallelCluster versão 3.5.0 e posterior para criar um cluster, e a criação de um cluster falhar com `--rollback-on-failure` definido como `false`, use o comando CLI [pcluster describe-cluster](#) para obter informações de status e falha. Nesse caso, o `clusterStatus` esperado da saída `pcluster describe-cluster` é `CREATE_FAILED`. Verifique a seção `failures` na saída para encontrar o `failureCode` e `failureReason`. Em seguida, na seção a seguir, encontre `failureCode` correspondente para obter ajuda adicional na solução de problemas. Para obter mais informações, consulte [pcluster describe-cluster](#).

Nas seções a seguir, recomendamos que você verifique os registros no nó principal, como os arquivos `/var/log/cfn-init.log` e `/var/log/chef-client.log`. Para obter mais informações sobre logs AWS ParallelCluster e como visualizá-los, consulte [Logs principais para depuração](#) e [Recuperando e preservando logs](#).

Se você não tiver um `failureCode`, navegue até o console AWS CloudFormation para ver a pilha de clusters. Verifique o `Status Reason` para ver se há `HeadNodeWaitCondition` ou falhas em outros recursos para encontrar detalhes adicionais da falha. Para obter mais informações, consulte [Veja os eventos AWS CloudFormation em CREATE_FAILED](#). Verifique os arquivos `/var/log/cfn-init.log` e `/var/log/chef-client.log` no nó principal.

failureCode é OnNodeConfiguredExecutionFailure

- Por que falhou?

Você forneceu um script personalizado na seção `OnNodeConfigured` do nó principal na configuração para criar um cluster. No entanto, o script personalizado falhou ao ser executado.

- Como resolver?

Verifique o arquivo `/var/log/cfn-init.log` para saber mais sobre a falha e como corrigir o problema em seu script personalizado. Perto do final desse log, você pode ver informações de execução relacionadas ao script `OnNodeConfigured` após a mensagem `Running command runpostinstall`.

failureCode é OnNodeConfiguredDownloadFailure

- Por que falhou?

Você forneceu um script personalizado na seção `OnNodeConfigured` do nó principal na configuração para criar um cluster. No entanto, o script personalizado falhou ao ser baixado..

- Como resolver?

Verifique se o URL é válido e se o acesso está configurado corretamente. Para obter mais informações sobre a configuração de scripts de bootstrap personalizados, consulte [Ações de bootstrap personalizadas](#).

Verifique o arquivo `/var/log/cfn-init.log`. Perto do final desse log, você pode ver informações de execução relacionadas ao processamento do script `OnNodeConfigured`, incluindo download, após a mensagem `Running command runpostinstall`.

failureCode é OnNodeConfiguredFailure

- Por que falhou?

Você forneceu um script personalizado na seção `OnNodeConfigured` do nó principal na configuração para criar um cluster. No entanto, o uso do script personalizado falhou na implantação do cluster. Uma causa imediata não pode ser determinada e é necessária uma investigação adicional.

- Como resolver?

Verifique o arquivo `/var/log/cfn-init.log`. Perto do final desse log, você pode ver informações de execução relacionadas ao processamento do script `OnNodeConfigured` após a mensagem `Running command runpostinstall`.

failureCode é OnNodeStartExecutionFailure

- Por que falhou?

Você forneceu um script personalizado na seção `OnNodeStart` do nó principal na configuração para criar um cluster. No entanto, o script personalizado falhou ao ser executado.

- Como resolver?

Verifique o arquivo `/var/log/cfn-init.log` para saber mais sobre a falha e como corrigir o problema em seu script personalizado. Perto do final desse log, você pode ver informações

de execução relacionadas ao script `OnNodeStart` após a mensagem `Running command runpreinstall`.

failureCode é OnNodeStartDownloadFailure

- Por que falhou?

Você forneceu um script personalizado na seção `OnNodeStart` do nó principal na configuração para criar um cluster. No entanto, o script personalizado falhou ao ser baixado..

- Como resolver?

Verifique se o URL é válido e se o acesso está configurado corretamente. Para obter mais informações sobre a configuração de scripts de bootstrap personalizados, consulte [Ações de bootstrap personalizadas](#).

Verifique o arquivo `/var/log/cfn-init.log`. Perto do final desse log, você pode ver informações de execução relacionadas ao processamento do script `OnNodeStart`, incluindo `download`, após a mensagem `Running command runpreinstall`.

failureCode é OnNodeStartFailure

- Por que falhou?

Você forneceu um script personalizado no `OnNodeStart` da seção do nó principal na configuração para criar um cluster. No entanto, o uso do script personalizado falhou na implantação do cluster. Uma causa imediata não pode ser determinada e é necessária uma investigação adicional.

- Como resolver?

Verifique o arquivo `/var/log/cfn-init.log`. Perto do final desse log, você pode ver informações de execução relacionadas ao processamento do script `OnNodeStart` após a mensagem `Running command runpreinstall`.

failureCode é EbsMountFailure

- Por que falhou?

Falha na montagem do volume do EBS definido na configuração do cluster.

- Como resolver?

Verifique o arquivo `/var/log/chef-client.log` para ver os detalhes da falha.

failureCode é EfsMountFailure

- Por que falhou?

Falha na montagem do volume do Amazon EFS definido na configuração do cluster.

- Como resolver?

Se você definiu um sistema de arquivos Amazon EFS existente, certifique-se de que o tráfego seja permitido entre o cluster e o sistema de arquivos. Para obter mais informações, consulte [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#).

Verifique o arquivo `/var/log/chef-client.log` para ver os detalhes da falha.

failureCode é FsxMountFailure

- Por que falhou?

O sistema de arquivos Amazon FSx definido na configuração do cluster falhou na montagem.

- Como resolver?

Se você definiu um sistema de arquivos Amazon FSx existente, certifique-se de que o tráfego seja permitido entre o cluster e o sistema de arquivos. Para obter mais informações, consulte [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#).

Verifique o arquivo `/var/log/chef-client.log` para ver os detalhes da falha.

failureCode é RaidMountFailure

- Por que falhou?

Falha na montagem dos volumes RAID definidos na configuração do cluster.

- Como resolver?

Verifique o arquivo `/var/log/chef-client.log` para ver os detalhes da falha.

failureCode é AmiVersionMismatch

- Por que falhou?

A versão AWS ParallelCluster usada para criar a AMI personalizada é diferente da versão AWS ParallelCluster usada para configurar o cluster. No CloudFormation console, visualize os detalhes da CloudFormation pilha de clusters e verifique o `Status Reason HeadNodeWaitCondition` para obter detalhes adicionais sobre AWS ParallelCluster as versões e a AMI. Para obter mais informações, consulte [Veja os eventos AWS CloudFormation em CREATE_FAILED](#).

- Como resolver?

A versão AWS ParallelCluster usada para criar a AMI personalizada deve ser a mesma da versão AWS ParallelCluster usada para configurar o cluster. Você pode alterar a versão personalizada da AMI ou a versão da CLI `pcluster` para torná-las iguais.

failureCode é InvalidAmi

- Por que falhou?

A AMI personalizada é inválida porque não foi criada usando o AWS ParallelCluster.

- Como resolver?

Use o comando `pcluster build-image` para criar uma AMI transformando sua AMI na imagem principal. Para obter mais informações, consulte [pcluster build-image](#).

failureCode é uma HeadNodeBootstrapFailure com failureReason Falha na configuração do nó principal.

- Por que falhou?

Uma causa imediata não pode ser determinada e é necessária uma investigação adicional. Por exemplo, pode ser que o cluster esteja em status protegido, e isso pode ser causado por uma falha no provisionamento da frota de computação estática.

- Como resolver?

Verifique o arquivo `/var/log/chef-client.log` para ver os detalhes da falha.

Note

Se você vir `RuntimeError` com uma exceção `Cluster state has been set to PROTECTED mode due to failures detected in static node provisioning`, o cluster está no status protegido. Para obter mais informações, consulte [Como depurar o modo protegido](#).

failureCode é uma **HeadNodeBootstrapFailure** com **failureReason** de tempo limite de criação do cluster.

- Por que falhou?

Por padrão, há um limite de tempo de 30 minutos para a conclusão da criação do cluster. Se a criação do cluster não for concluída dentro desse período, a criação do cluster falhará com um erro de tempo limite. A criação do cluster pode atingir o tempo limite por diferentes motivos. Por exemplo, falhas de tempo limite podem ser causadas por uma falha na criação do nó principal, um problema de rede, scripts personalizados que demoram muito para serem executados no nó principal, um erro em um script personalizado executado nos nós de computação ou longos tempos de espera para o provisionamento do nó de computação. Uma causa imediata não pode ser determinada e é necessária uma investigação adicional.

- Como resolver?

Verifique os arquivos `/var/log/cfn-init.log` e `/var/log/chef-client.log` para ver os detalhes da falha. Para obter mais informações sobre logs AWS ParallelCluster e como obtê-los, consulte [Logs principais para depuração](#) e [Recuperando e preservando logs](#).

Você pode descobrir o seguinte nesses logs.

- Vendo **Waiting for static fleet capacity provisioning** perto do final do **chef-client.log**

Isso indica que a criação do cluster atingiu o tempo limite ao aguardar a inicialização dos nós estáticos. Para obter mais informações, consulte [Vendo erros nas inicializações dos nós de computação](#).

- A visualização do script do nó principal **OnNodeConfigured** ou **OnNodeStart** não foi concluído no final do **cfn-init.log**

Isso indica que o script personalizado **OnNodeConfigured** ou o **OnNodeStart** demorou muito para ser executado e causou um erro de tempo limite. Verifique se há problemas no script personalizado que podem fazer com que ele seja executado por um longo tempo. Se o script personalizado exigir muito tempo para ser executado, considere alterar o limite de tempo limite adicionando uma seção **DevSettings** ao arquivo de configuração do cluster, conforme mostrado no exemplo a seguir:

```
DevSettings:
  Timeouts:
    HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- Não consigo encontrar os logs ou o nó principal não foi criado com sucesso

É possível que o nó principal não tenha sido criado com sucesso e que os logs não possam ser encontrados. No CloudFormation console, visualize os detalhes da pilha do cluster para verificar detalhes adicionais da falha.

failureCode é uma **HeadNodeBootstrapFailure** com **failureReason** de Falha no bootstrap do nó principal.

- Por que falhou?

Uma causa imediata não pode ser determinada e é necessária uma investigação adicional.

- Como resolver?

Verifique os arquivos `/var/log/cfn-init.log` e `/var/log/chef-client.log`.

failureCode é **ResourceCreationFailure**

- Por que falhou?

A criação de alguns recursos falhou durante o processo de criação do cluster. A falha pode ocorrer por vários motivos. Por exemplo, falhas na criação de recursos podem ser causadas por problemas de capacidade ou por uma política de IAM mal configurada.

- Como resolver?

No CloudFormation console, visualize a pilha do cluster para verificar detalhes adicionais da falha na criação de recursos.

failureCode é ClusterCreationFailure

- Por que falhou?

Uma causa imediata não pode ser determinada e é necessária uma investigação adicional.

- Como resolver?

No CloudFormation console, visualize a pilha do cluster e verifique a Status Reason HeadNodeWaitCondition para encontrar detalhes adicionais da falha.

Verifique os arquivos `/var/log/cfn-init.log` e `/var/log/chef-client.log`.

Vendo WaitCondition timed out... na CloudFormation pilha

Para obter mais informações, consulte [failureCode é uma HeadNodeBootstrapFailure com failureReason de tempo limite de criação do cluster..](#)

Vendo Resource creation cancelled na CloudFormation pilha

Para obter mais informações, consulte [failureCode é ResourceCreationFailure.](#)

Visualizando Failed to run cfn-init... ou outros erros na pilha AWS CloudFormation

Verifique `/var/log/cfn-init.log` e `/var/log/chef-client.log` para ver os detalhes adicionais da falha.

Visualizando chef-client.log que termina com INFO: Waiting for static fleet capacity provisioning

Isso está relacionado ao tempo limite de criação do cluster ao aguardar a inicialização dos nós estáticos. Para obter mais informações, consulte [Vendo erros nas inicializações dos nós de computação.](#)

Vendo **Failed to run preinstall or postinstall in cfn-init.log**

Você tem um script `OnNodeConfigured` ou `OnNodeStart` na seção `HeadNode` de configuração do cluster. O script não está funcionando corretamente. Verifique o arquivo `/var/log/cfn-init.log` para ver os detalhes do erro do script personalizado.

Vendo **This AMI was created with xxx, but is trying to be used with xxx...** na CloudFormation pilha

Para obter mais informações, consulte [failureCode é AmiVersionMismatch](#).

Vendo **This AMI was not baked by AWS ParallelCluster...** na CloudFormation pilha

Para obter mais informações, consulte [failureCode é InvalidAmi](#).

Vendo que o comando **pcluster create-cluster** falha ao ser executado localmente

Verifique o `~/parallelcluster/pcluster-cli.log` em seu sistema de arquivos local para ver os detalhes da falha.

Suporte adicional

Siga as orientações de solução de problemas em [Solução de problemas de implantação de cluster](#).

Verifique se seu cenário está abordado em [Problemas GitHub conhecidos](#) em AWS ParallelCluster on GitHub.

Para obter suporte adicional, consulte [Suporte adicional](#).

Tentar executar um trabalho

O trabalho interativo do `srun` falha com erro **`srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf`**

- Por que falhou?

Você executou o comando `srun` para enviar um trabalho e, em seguida, aumentou o tamanho de uma fila usando o comando `pcluster update-cluster`, sem reiniciar os daemons Slurm após a conclusão da atualização.

O Slurm organiza os daemons Slurm em uma hierarquia de árvores para otimizar a comunicação. Essa hierarquia só é atualizada quando os daemons são iniciados.

Suponha que você use `srun` para iniciar um trabalho e, em seguida, executar o comando `pcluster update-cluster` para aumentar o tamanho da fila. Novos nós de computação são lançados como parte da atualização. Em seguida, o Slurm enfileira seu trabalho em um dos novos nós de computação. Nesse caso, tanto os daemons Slurm quanto o `srun` não detectam os novos nós de computação. O `srun` retorna um erro porque não detecta os novos nós.

- Como resolver?

Reinicie os daemons Slurm em todos os nós de computação e use `srun` para enviar seu trabalho. Você pode programar a reinicialização dos daemons Slurm executando o comando `scontrol reboot` que reinicia os nós de computação. Para obter mais informações, consulte [scontrol reboot](#) na documentação do Slurm. Você também pode reiniciar manualmente os daemons Slurm nos nós de computação solicitando a reinicialização dos serviços `systemd` correspondentes.

Trabalho está preso no estado **CF** com o comando **squeue**

Isso pode ser um problema com a inicialização dos nós dinâmicos. Para obter mais informações, consulte [Vendo erros nas inicializações dos nós de computação](#).

Executando trabalhos em grande escala e vendo **nfsd: too many open connections, consider increasing the number of threads in /var/log/messages**

Com um sistema de arquivos em rede, quando os limites da rede são atingidos, o tempo de espera de E/S também aumenta. Isso pode resultar em travamentos leves porque a rede é usada para gravar dados para métricas de rede e de E/S.

Com instâncias de 5ª geração, usamos o driver ENA para expor os contadores de pacotes. Esses contadores contam os pacotes moldados pelo AWS em que a rede atinge os limites de largura de banda da instância. Você pode verificar esses contadores para ver se eles são maiores que 0. Se estiverem, você excedeu seus limites de largura de banda. Você pode ver esses contadores executando `ethtool -S eth0 | grep exceeded`.

Exceder os limites da rede geralmente é resultado do suporte a muitas conexões NFS. Essa é uma das primeiras coisas a verificar quando você atinge ou excede os limites da rede.

Por exemplo, a saída a seguir mostra pacotes descartados:

```
$ ethtool -S eth0 | grep exceeded
bw_in_allowance_exceeded: 38750610
bw_out_allowance_exceeded: 1165693
pps_allowance_exceeded: 103
contrack_allowance_exceeded: 0
linklocal_allowance_exceeded: 0
```

Para evitar receber essa mensagem, considere alterar o tipo de instância do nó principal para um tipo de instância com melhor desempenho. Considere transferir seu armazenamento de dados para sistemas de arquivos de armazenamento compartilhado que não sejam exportados como um compartilhamento NFS, como Amazon EFS ou Amazon FSx. Para obter mais informações, consulte [Armazenamento compartilhado](#) e as [melhores práticas](#) no AWS ParallelCluster Wiki em GitHub.

Executando um trabalho de MPI

Habilitar o modo de depuração

Para ativar o modo de depuração do OpenMPI, consulte [What controls does Open MPI have that aid in debugging \(Quais controles o Open MPI tem que auxiliam na depuração\)](#).

Para ativar o modo de depuração do IntelMPI, consulte [Other Environment Variables \(Outras variáveis de ambiente\)](#).

Visualizando **MPI_ERRORS_ARE_FATAL** e **OPAL ERROR** na saída do trabalho

Esses códigos de erro vêm da camada MPI em seu aplicativo. Para saber como obter logs de depuração do MPI do seu aplicativo, consulte [Habilitar o modo de depuração](#).

Uma possível causa desse erro é que seu aplicativo foi compilado para uma implementação de MPI específica, como o OpenMPI, e você está tentando executá-lo com uma implementação de MPI diferente, como o IntelMPI. Verifique se você está compilando e executando seu aplicativo com a mesma implementação de MPI.

Usando **mpirun** com o DNS gerenciado desativado

Para clusters criados com [SlurmSettings/Dns/DisableManagedDns](#) e [UseEc2Hostnames](#) definidos como `true`, o nome do Slurm nó não é resolvido pelo DNS. Slurmpodem inicializar processos MPI quando nodenames não estão habilitados e se a tarefa MPI for executada em um contexto. Slurm Recomendamos seguir as orientações do [Guia do Usuário de MPI do Slurm](#) para executar trabalhos do MPI com Slurm.

Tentando atualizar um cluster

O comando **pcluster update-cluster** falha ao ser executado localmente

Verifique o `~/parallelcluster/pcluster-cli.log` em seu sistema de arquivos local para ver os detalhes da falha.

Ver **clusterStatus** é **UPDATE_FAILED** com comando **pcluster describe-cluster**

Se a atualização da pilha de clusters for revertida, verifique os detalhes do erro no arquivo `/var/log/chef-client.logs`.

Verifique se seu problema foi mencionado em [Problemas GitHub conhecidos](#) em AWS ParallelCluster on GitHub.

A atualização do cluster atingiu o tempo limite

Isso pode ser um problema relacionado ao `cf-n-hup` não sendo executado. Se o daemon `cf-n-hup` for encerrado por uma causa externa, ele não será reiniciado automaticamente. Se `cf-n-hup` não estiver em execução, durante uma atualização do cluster, a CloudFormation pilha inicia o processo de atualização conforme o esperado, mas o procedimento de atualização não é ativado no nó principal e a implantação da pilha eventualmente expira. Para obter mais informações, consulte [Solução de problemas de tempo limite de atualização de cluster quando `cf-n-hup` não está em execução](#) para solucionar o problema e corrigir o problema.

Tentar acessar o armazenamento

Usar um sistema de arquivos externo do Amazon FSx for Lustre

Certifique-se de que o tráfego seja permitido entre o cluster e o sistema de arquivos. Se estiver usando um sistema de arquivos existente, ele deve ser associado a um grupo de segurança que permite o tráfego de entrada e saída do TCP pelas portas 988, 1021, 1022, e 1023. Para obter mais informações sobre como configurar grupos de segurança, consulte [FileSystemId](#).

Usar um sistema de arquivos externo do Amazon Elastic File System

Certifique-se de que o tráfego seja permitido entre o cluster e o sistema de arquivos. Se estiver usando um sistema de arquivos existente, ele deve ser associado a um grupo de segurança que permite o tráfego de entrada e saída do TCP pelas portas 988, 1021, 1022, e 1023. Para obter mais informações sobre como configurar grupos de segurança, consulte [FileSystemId](#).

Tentando excluir um cluster

O comando `pcluster delete-cluster` falha ao ser executado localmente

Verifique o arquivo `~/parallelcluster/pcluster-cli.log` em seu sistema de arquivos local.

A pilha de clusters não consegue ser excluída

Se a pilha do cluster não for excluída, verifique a mensagem de eventos da CloudFormation pilha.

Verifique se seu problema está mencionado em [Problemas GitHub conhecidos](#) em AWS ParallelCluster on GitHub.

Tentando atualizar a pilha de APIs AWS ParallelCluster

Verifique se seu problema está mencionado em [Problemas GitHub conhecidos](#) em AWS ParallelCluster on GitHub.

Vendo erros nas inicializações dos nós de computação

Vendo **Node bootstrap error** em **clustermgtd.log**

O problema está relacionado à falha na inicialização dos nós de computação. Para obter informações sobre como depurar um problema no modo protegido por cluster, consulte [Como depurar o modo protegido](#).

Eu configurei reservas de capacidade sob demanda (ODCRs) ou instâncias reservadas zonais

ODCRs que incluem instâncias que têm várias interfaces de rede, como P4d, P4de, e AWS Trainium (Trn)

No arquivo de configuração do cluster, verifique se o HeadNode está em uma sub-rede pública e se os nós de computação estão em uma sub-rede privada.

Os ODCRs são ODCRs direcionados

Vendo, **Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'**. embora eu já tenha **/opt/slurm/etc/pcluster/run_instances_overrides.json** instalado, seguindo as instruções dadas em [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#)

Se você estiver usando AWS ParallelCluster, versões 3.1.1 a 3.2.1, com ODCRs direcionados e também estiver usando o [arquivo JSON de substituição de instâncias de execução](#), é possível que você não tenha o arquivo JSON formatado corretamente. Você pode ver um erro em `clustermgtd.log`, como o seguinte:

```
Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'.  
Using default: {} in /var/log/parallelcluster/clustermgtd.
```

Valide se o formato de arquivo JSON está correto executando o seguinte:

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

Ver **Found RunInstances parameters override.** em **clustermgtd.log** quando a criação do cluster falhou ou em **slurm_resume.log** quando o trabalho de execução falhou

Se você estiver usando o [arquivo JSON de substituição de instâncias de execução](#), verifique se definiu corretamente o nome da fila e o nome dos recursos de computação no arquivo `/opt/slurm/etc/pcluster/run_instances_overrides.json`.

Ver **An error occurred (InsufficientInstanceCapacity)** no **slurm_resume.log** quando não conseguiu executar um trabalho ou em **clustermgtd.log** quando eu não consigo criar um cluster

Usando PG-ODCR (grupo de posicionamento ODCR)

Ao criar um ODCR com um grupo de posicionamento associado, o mesmo nome do grupo de posicionamento deve ser usado no arquivo de configuração. Defina o [nome do grupo de posicionamento](#) correspondente na configuração do cluster.

Usar instâncias reservadas zonais

Se você estiver usando instâncias reservadas zonais com PlacementGroup / Enabled para true na configuração do cluster, poderá ver um erro, como o seguinte:

```
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.
```

Você pode vê-lo porque as instâncias reservadas zonais não são colocadas na mesma UC (ou coluna), o que pode causar erros de capacidade insuficientes (ICEs) ao usar grupos de posicionamento. Você pode verificar esse caso desativando a configuração de

GrupoPlacementGroup na configuração do cluster para determinar se o cluster pode alocar as instâncias.

Ver **An error occurred (VcpuLimitExceeded)** no **slurm_resume.log** quando não consegui executar um trabalho ou em **clustermgtd.log** quando eu não consigo criar um cluster

Verifique os limites de vCPU em sua conta para o tipo específico de instância EC2 que você está usando. Se você ver zero ou menos vCPUs do que está solicitando, solicite um aumento para seus limites. Para obter informações sobre como visualizar limites atuais e solicitar novos limites, consulte [Service Quotas do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Ver **An error occurred (InsufficientInstanceCapacity)** no **slurm_resume.log** quando não consegui executar um trabalho ou em **clustermgtd.log** quando eu não consigo criar um cluster

Você está enfrentando um problema de capacidade insuficiente. Siga <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> para solucionar o problema.

Vendo que os nós estão em estado **DOWN** com **Reason (Code:InsufficientInstanceCapacity)...**

Você está enfrentando um problema de capacidade insuficiente. Siga <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> para solucionar o problema. Para obter mais informações sobre o modo de failover rápido de capacidade insuficiente do AWS ParallelCluster, consulte [Failover rápido de capacidade insuficiente do cluster Slurm](#).

Vendo **cannot change locale (en_US.utf-8) because it has an invalid name** em **slurm_resume.log**

Isso pode ocorrer se você tiver um processo de instalação do yum malsucedido que deixou as configurações de localidade em um estado inconsistente. Por exemplo, isso pode ser causado quando um usuário encerra o processo de instalação.

Para verificar a causa, realize as seguintes ações:

- Executar `su - pcluster-admin`.

O shell mostra um erro, como `cannot change locale...no such file or directory`.

- Executar `localedef --list`.

Retorna uma lista vazia ou não contém a localidade padrão.

- Verifique o último comando yum com `yum history` e `yum history info #ID`. O último ID tem `Return-Code: Success?`

Se a última ID não tiver `Return-Code: Success`, os scripts de pós-instalação podem não ter sido executados com êxito.

Para corrigir o problema, tente reconstruir a localidade com `yum reinstall glibc-all-langpacks`. Após a reconstrução, `su - pcluster-admin` não mostra um erro ou aviso se o problema foi corrigido.

Nenhum dos cenários anteriores se aplica à minha situação

Para solucionar problemas de inicialização do nó de computação, consulte [Solução de problemas de inicialização do nó](#).

Verifique se seu cenário está abordado em [Problemas GitHub conhecidos](#) em AWS ParallelCluster on GitHub.

Para obter suporte adicional, consulte [Suporte adicional](#).

Métricas de integridade do cluster para solução de problemas

As métricas de integridade do cluster são adicionadas ao painel do Amazon CloudWatch AWS ParallelCluster a partir da AWS ParallelCluster versão 3.6.0. Nas seções a seguir, você vai aprender sobre as métricas de integridade do painel e sobre ações que você pode realizar para solucionar problemas.

Tópicos

- [Visualizando o gráfico de Instance Provisioning Errors \(Erros de provisionamento de instâncias\)](#)
- [Visualizando o gráfico de Unhealthy Instance Errors \(Erros de instância não saudáveis\)](#)
- [Visualizando o gráfico de Tempo de inatividade da frota de computadores](#)

Visualizando o gráfico de Instance Provisioning Errors (Erros de provisionamento de instâncias)

Se você ver um valor diferente de zero no gráfico Instance Provisioning Errors, isso significa que a instância do EC2 para apoiar os nós do Slurm falhou ao iniciar na API CreateFleet ou RunInstance.

Vendo **IAMPolicyErrors**

- O que aconteceu?

Várias instâncias falharam na inicialização, o que é causado por permissões insuficientes com código de erro. `UnauthorizedOperation`

- Como resolver?

Se você configurou um [InstanceRole](#) ou [InstanceProfile](#) personalizado, verifique suas políticas do IAM e verifique se está usando as credenciais corretas.

Verifique o arquivo `clustermgtd` para ver os detalhes do erro do nó estático. Verifique o arquivo `slurm_resume.log` para ver os detalhes do erro do nó dinâmico. Use os detalhes para saber mais sobre as permissões ausentes que devem ser adicionadas.

Vendo **VcpuLimitErrors**

- O que aconteceu?

O AWS ParallelCluster falhou ao iniciar instâncias porque atingiu o limite de vCPU na sua Conta da AWS para um tipo específico de instância EC2 que você configurou para nós de computação de cluster.

- Como resolver?

Verifique o erro `VcpuLimitExceeded` no arquivo `clustermgtd` para nós estáticos e verifique se há nós dinâmicos no arquivo `slurm_resume.log` para obter detalhes adicionais. Para resolver esse problema, é possível solicitar um aumento nos limites da vCPU. Para obter mais informações sobre como visualizar limites atuais e solicitar novos limites, consulte [Service Quotas do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Vendo **VolumeLimitErrors**

- O que aconteceu?

Você atingiu o limite de volume do Amazon EBS no seu Conta da AWS, e o AWS ParallelCluster não consegue iniciar instâncias com código de erro `InsufficientVolumeCapacity` ou `VolumeLimitExceeded`.

- Como resolver?

Verifique o arquivo `clustermgtd` para ver se há nós estáticos, e verifique se há nós dinâmicos no arquivo `slurm_resume.log` para obter detalhes adicionais sobre limite de volume. Para resolver esse problema, você pode usar um outro Região da AWS, limpar os volumes existentes ou entrar em contato com o Support Center da AWS para enviar uma solicitação e aumentar seu limite de volume do Amazon EBS.

Vendo **InsufficientCapacityErrors**

- O que aconteceu?

O AWS ParallelCluster não tem capacidade suficiente para iniciar instâncias do EC2 nos nós traseiros.

- Como resolver?

Verifique se há nós estáticos no arquivo `clustermgtd` e verifique se há nós dinâmicos no arquivo `slurm_resume.log` para obter detalhes de erro de capacidade insuficientes. Para solucionar o problema, siga as orientações em <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>.

OtherInstanceLaunchFailures

- O que aconteceu?

A instância do EC2 para suportar os nós de computação falhou ao ser iniciada com a API `CreateFleet` ou `RunInstance`.

- Como resolver?

Verifique se há nós estáticos no arquivo `clustermgtd` e verifique se há nós dinâmicos no arquivo `slurm_resume.log` para obter detalhes do erro.

Visualizando o gráfico de Unhealthy Instance Errors (Erros de instância não saudáveis)

- O que aconteceu?

Várias instâncias de computação foram iniciadas, mas depois encerradas por não serem íntegras.

- Como resolver?

Para obter mais informações sobre solução de problemas de nós não saudáveis, consulte [Solução de problemas inesperados de substituições e encerramentos de nós](#).

Vendo **InstanceBootstrapTimeoutError**

- O que aconteceu?

Uma instância não pode se juntar ao cluster em `resume_timeout` (para nós dinâmicos) ou `node_replacement_timeout` (para nós estáticos). Isso pode ocorrer se a rede não estiver configurada corretamente para os nós de computação, ou se os scripts personalizados executados no nó de computação demorarem muito para serem concluídos.

- Como resolver?

Para nós dinâmicos, verifique no log `clustermgtd (/var/log/parallelcluster/clustermgtd)` o endereço IP do nó de computação e erros como os seguintes:

```
Node bootstrap error: Resume timeout expires for node
```

Para nós estáticos, verifique no log `clustermgtd (/var/log/parallelcluster/clustermgtd)` o endereço IP do nó de computação e erros como os seguintes:

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

Para obter detalhes adicionais, verifique se há erros no arquivo `/var/log/cloud-init-output.log`. Você pode recuperar endereços IP de nós de computação problemáticos a partir dos arquivos de log `clustermgtd` e `slurm_resume`.

Vendo **EC2HealthCheckErrors**

- O que aconteceu?

Uma instância falhou em uma verificação de integridade do EC2.

- Como resolver?

Para obter informações sobre como solucionar esse problema, consulte [Solução de problemas em instâncias com falha nas verificações de status](#).

Vendo **ScheduledEventHealthCheckErrors**

- O que aconteceu?

Uma instância falhou em uma verificação de integridade de um evento agendado pelo EC2 e não está íntegra.

- Como resolver?

Para obter informações sobre como solucionar esse problema, consulte [Eventos programados para instâncias](#).

Vendo **NoCorrespondingInstanceErrors**

- O que aconteceu?

O AWS ParallelCluster não consegue encontrar nós de apoio a instâncias. Os nós provavelmente terminaram automaticamente durante as operações de bootstrap. scripts [SlurmQueues](#) / [CustomActions](#) / [OnNodeStart](#) | [OnNodeConfigured](#) ou erros de rede podem produzir `NoCorrespondingInstanceErrors`.

- Como resolver?

Para obter detalhes adicionais, consulte `/var/log/cloud-init-output.log` para ver o nó de computação.

Visualizando o gráfico de Tempo de inatividade da frota de computadores

Observando um **MaxDynamicNodeIdleTime** que é significativamente maior do que o limite de redução do tempo de inatividade

- O que aconteceu?

Sua instância não está sendo encerrada corretamente. O `MaxDynamicNodeIdleTime` mostra o tempo máximo em segundos em que um nó dinâmico, apoiado por uma instância do EC2, fica ocioso. O limite de redução do tempo de inatividade é derivado do parâmetro [ScaledownIdleTime](#) de configuração do cluster. Quando um nó de computação fica ocioso por mais segundos do que o tempo de redução do tempo de inatividade, o Slurm desliga o nó e AWS ParallelCluster encerra a instância de backup. Nesse caso, algo está impedindo o encerramento da instância.

- Como resolver?

Para obter mais informações sobre esse problema, consulte [Substituindo, encerrando ou desligando instâncias e nós problemáticos](#) em [Solucionar problemas de escala](#).

Solução de problemas de implantação de cluster

Se o cluster não for criado e reverter a criação da pilha, você poderá examinar os arquivos de log para diagnosticar o problema. A mensagem de falha provavelmente se parece com a seguinte saída:

```
$ 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": "3.7.0",
    "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"
}
```

Tópicos

- [Veja os eventos AWS CloudFormation em CREATE_FAILED](#)
- [Use a CLI para visualizar fluxos de log](#)
- [Recrie o cluster com falha com rollback-on-failure](#)

Veja os eventos AWS CloudFormation em **CREATE_FAILED**

Você pode usar o console ou a AWS ParallelCluster CLI para visualizar CloudFormation eventos sobre CREATE_FAILED erros e ajudar a encontrar a causa raiz.

Tópicos

- [Exibir eventos no CloudFormation console](#)
- [Use a CLI para visualizar e filtrar CloudFormation eventos em CREATE_FAILED](#)

Exibir eventos no CloudFormation console

Para ver mais informações sobre o que causou o "CREATE_FAILED" status, você pode usar o CloudFormation console.

Veja as mensagens de CloudFormation erro do console.

1. Faça login no AWS Management Console, e acesse <https://console.aws.amazon.com/cloudformation>.
2. Selecione a pilha chamada *cluster_name*.
3. Escolha a guia Eventos.

4. Verifique o status do recurso que não foi criado percorrendo a lista de eventos do recurso por Logical ID (ID lógico). Se houver falha na criação de uma subtarefa, retroceda para encontrar o evento de recurso com falha.
5. Por exemplo, se você ver a mensagem de status a seguir, deverá usar tipos de instância que não excedam seu limite atual de vCPU ou solicitar mais capacidade de 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).
```

Use a CLI para visualizar e filtrar CloudFormation eventos em **CREATE_FAILED**

Para diagnosticar o problema de criação do cluster, você pode usar o comando [pcluster get-cluster-stack-events](#) filtrando por status `CREATE_FAILED`. Para obter mais informações, consulte [Filtragem de saída AWS CLI](#) no Guia de usuário AWS Command Line Interface.

```
$ 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"
  }
]

```

No exemplo anterior, a falha estava na configuração do nó principal.

Use a CLI para visualizar fluxos de log

Para depurar esse tipo de problema, você pode listar os fluxos de log disponíveis no nó principal com o [pcluster list-cluster-log-streams](#), filtrando `node-type` e analisando o conteúdo dos fluxos de 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",

```

```

    ...
  },
  ...
]
}

```

Os dois fluxos de log principais que você pode usar para encontrar erros de inicialização são os seguintes:

- `cfn-init` é o log do script `cfn-init`. Primeiro, verifique esse fluxo de logs. É provável que você veja o erro `Command chef failed` nesse log. Veja as linhas imediatamente antes dessa linha para obter mais detalhes relacionados à mensagem de erro. Para mais informações, consulte [cfn-init](#).
- `cloud-init` é o log do [cloud-init](#). Se não vir nada no `cfn-init`, tente verificar esse log a seguir.

Você pode recuperar o conteúdo do fluxo de log usando o [pcluster get-cluster-log-events](#) (observe a opção `--limit 5` de limitar o número de eventos recuperados):

```

$ 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\"
  }
]
}
```

No exemplo anterior, a falha é causada por uma falha `runpostinstall`, portanto, está estritamente relacionada ao conteúdo do script de bootstrap personalizado usado no parâmetro de configuração `OnNodeConfigured` do [CustomActions](#).

Recrie o cluster com falha com **rollback-on-failure**

AWS ParallelCluster cria fluxos de CloudWatch log de cluster em grupos de registros. Você pode visualizar esses registros no CloudWatch console Painéis personalizados ou grupos de registros. Para obter mais informações, consulte [Painel do Amazon CloudWatch](#) e [Integração com Amazon CloudWatch Logs](#). Se não houver fluxos de log disponíveis, a falha pode ser causada pelo script de bootstrap [CustomActions](#) personalizado ou por um problema relacionado à AMI. Para diagnosticar o problema de criação nesse caso, crie o cluster novamente usando [pcluster create-cluster](#), incluindo o parâmetro `--rollback-on-failure` definido como `false`. Em seguida, use o SSH para visualizar o cluster, conforme mostrado a seguir:

```
$ 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": "3.7.0",
"clusterStatus": "CREATE_IN_PROGRESS"
}
}
$ pcluster ssh --cluster-name mycluster
```

Depois de fazer login no nó principal, você deve encontrar três arquivos de log principais que podem ser usados para encontrar o erro.

- `/var/log/cfn-init.log` é o log do script `cfn-init`. Primeiro, verifique esse log. É provável que você veja o erro como o `Command chef failed` nesse log. Veja as linhas imediatamente antes dessa linha para obter mais detalhes relacionados à mensagem de erro. Para mais informações, consulte [cfn-init](#).
- `/var/log/cloud-init.log` é o log do [cloud-init](#). Se não vir nada no `cfn-init.log`, tente verificar esse log a seguir.
- `/var/log/cloud-init-output.log` é a saída dos comandos que foram executados pelo [cloud-init](#). Isso inclui a saída de `cfn-init`. Na maioria dos casos, não é preciso consultar esse log para solucionar esse tipo de problema.

Solucionar problemas de escala

Esta seção é relevante para clusters que foram instalados usando o AWS ParallelCluster versão 3.0.0 e posterior com o programador de tarefas do Slurm. Para obter mais informações sobre a configuração múltiplas filas, consulte [Configuração de várias filas](#).

Se um de seus clusters em execução estiver enfrentando problemas, coloque o cluster em um estado STOPPED executando o comando a seguir antes de começar a solucionar o problema. Isso evita incorrer em custos inesperados.

```
$ pcluster update-compute-fleet --cluster-name mycluster \  
--status STOP_REQUESTED
```

Você pode listar os fluxos de log disponíveis nos nós do cluster usando o comando [pcluster list-cluster-log-streams](#) e filtrando por meio do `private-dns-name` de um dos nós com falha ou o nó principal:

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \  
--filters 'Name=private-dns-name,Values=ip-10-0-0-101'
```

Em seguida, você pode recuperar o conteúdo do fluxo de logs para analisá-lo usando o comando [pcluster get-cluster-log-events](#) e transmitindo o `--log-stream-name` correspondente a um dos principais logs mencionados na seção a seguir:

```
$ 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 cria fluxos de CloudWatch log de cluster em grupos de registros. Você pode visualizar esses registros no CloudWatch console Painéis personalizados ou grupos de registros. Para obter mais informações, consulte [Painel do Amazon CloudWatch](#) e [Integração com Amazon CloudWatch Logs](#).

Tópicos

- [Logs principais para depuração](#)
- [Ver o erro `InsufficientInstanceCapacity` no `slurm_resume.log` quando não consegui executar um trabalho ou em `clustermgtd.log` quando eu não consigo criar um cluster](#)
- [Solução de problemas de inicialização do nó](#)
- [Solução de problemas inesperados de substituições e encerramentos de nós](#)
- [Substituindo, encerrando ou desligando instâncias e nós problemáticos](#)
- [Status Inactive da fila \(partição\)](#)
- [Solucionando outros problemas conhecidos de nós e tarefas](#)

Logs principais para depuração

A tabela a seguir fornece uma visão geral dos principais logs do nó principal:

- `/var/log/cfn-init.log` - Esse é o log AWS CloudFormation init. Ele contém todos os comandos que foram executados quando uma instância foi configurada. Use-o para solucionar problemas de inicialização.

- `/var/log/chef-client.log` - Este é o log do cliente Chef. Ele contém todos os comandos que foram executados pelo Chef/CINC. Use-o para solucionar problemas de inicialização.
- `/var/log/parallelcluster/slurm_resume.log` - Isso é um log ResumeProgram. Ele lança instâncias para nós dinâmicos. Use-o para solucionar problemas de inicialização de nós dinâmicos.
- `/var/log/parallelcluster/slurm_suspend.log` - Esse é o log SuspendProgram. É chamado quando as instâncias são encerradas para nós dinâmicos. Use-o para solucionar problemas de encerramento de nós dinâmicos. Ao verificar esse registro, você também deve verificar o log `clustermgtd`.
- `/var/log/parallelcluster/clustermgtd` - Esse é o log `clustermgtd`. Ele é executado como o daemon centralizado que gerencia a maioria das ações de operação do cluster. Use-o para solucionar qualquer problema de inicialização, encerramento ou operação do cluster.
- `/var/log/slurmctld.log` - Este é o log do daemon de controle do Slurm. AWS ParallelCluster não toma decisões de escala. Em vez disso, ele apenas tenta lançar recursos para satisfazer os requisitos do Slurm. É útil para problemas de escalab e alocação, problemas relacionados ao trabalho e quaisquer problemas de lançamento e encerramento relacionados ao programador.
- `/var/log/parallelcluster/compute_console_output` - Esse log registra a saída do console de um subconjunto de amostra de nós de computação estáticos que foram encerrados inesperadamente. Use esse registro se os nós de computação estáticos terminarem e os registros dos nós de computação não estiverem disponíveis em CloudWatch O conteúdo `compute_console_output` log que você recebe é o mesmo quando você usa o console do EC2 ou AWS CLI para recuperar a saída do console da instância.

Estes são os principais logs dos nós de computação:

- `/var/log/cloud-init-output.log` - Esse é o log [cloud-init](#). Ele contém todos os comandos que foram executados quando uma instância foi configurada. Use-o para solucionar problemas de inicialização.
- `/var/log/parallelcluster/computemgtd` - Esse é o log `computemgtd`. Ele é executado em cada nó de computação para monitorar o nó no caso incomum de o daemon `clustermgtd` no nó principal estar off-line. Use-o para solucionar problemas inesperados de encerramento.
- `/var/log/slurmd.log` - Este é o log do daemon de computação do Slurm. Use-o para solucionar problemas de inicialização e de falhas de computação.

Ver o erro **InsufficientInstanceCapacity** no **slurm_resume.log** quando não consegui executar um trabalho ou em **clustermgtd.log** quando eu não consigo criar um cluster

Se o cluster usa um programador Slurm, você está enfrentando um problema de capacidade insuficiente. Se não houver instâncias suficientes disponíveis quando a solicitação de inicialização de instância é feita, um erro `InsufficientInstanceCapacity` será gerado.

Para a capacidade estática da instância, você pode encontrar o erro no log `clustermgtd` em `/var/log/parallelcluster/clustermgtd`.

Para a capacidade dinâmica da instância, você pode encontrar o erro no log `ResumeProgram` em `/var/log/parallelcluster/slurm_resume.log`.

A mensagem é parecida ao seguinte exemplo:

```
An error occurred (InsufficientInstanceCapacity) when calling the RunInstances/
CreateFleet operation...
```

Com base no seu caso de uso, considere usar um dos seguintes métodos para evitar receber esses tipos de mensagens de erro:

- Desative o grupo de posicionamento se ele estiver ativado. Para obter mais informações, consulte [Grupos de posicionamento e problemas de execução de instâncias](#).
- Reserve capacidade para as instâncias e inicie-as com o ODCR (On-Demand Capacity Reservations). Para obter mais informações, consulte [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#).
- Configure diversos recursos computacionais com diferentes tipos de instância. Se sua carga de trabalho não exigir um tipo de instância específico, você pode aproveitar o failover rápido de capacidade insuficiente com vários recursos computacionais. Para obter mais informações, consulte [Failover rápido de capacidade insuficiente do cluster Slurm](#).
- Configure vários tipos de instância no mesmo recurso computacional e aproveite a alocação de vários tipos de instância. Para obter mais informações sobre como configurar várias instâncias, consulte [Alocação a vários tipos de instância com o Slurm](#) e [Scheduling / SlurmQueues / ComputeResources / Instances](#).
- Mova a fila para uma zona de disponibilidade diferente alterando o ID da sub-rede na configuração do cluster [Scheduling / SlurmQueues / Networking / SubnetIds](#).

- Se sua workload não estiver fortemente acoplada, divida a fila em diferentes zonas de disponibilidade. Para obter mais informações sobre como configurar várias sub-redes, consulte [Scheduling](#) / [SlurmQueues](#) / [Networking](#) / [SubnetIds](#).

Solução de problemas de inicialização do nó

Esta seção aborda como você pode solucionar problemas de inicialização do nó. Isso inclui problemas em que o nó falha ao iniciar, ligar ou ingressar em um cluster.

Tópicos

- [Nó principal](#)
- [Nós de computação](#)

Nó principal

Logs aplicáveis:

- `/var/log/cfn-init.log`
- `/var/log/chef-client.log`
- `/var/log/parallelcluster/clustermgtd`
- `/var/log/parallelcluster/slurm_resume.log`
- `/var/log/slurmctld.log`

Verifique os logs `/var/log/cfn-init.log` e `/var/log/chef-client.log` ou os fluxos de log correspondentes. Esses logs contêm todas as ações que foram executadas quando o nó principal foi configurado. A maioria dos erros que ocorrem durante a configuração deve ter mensagens de erro localizadas no log `/var/log/chef-client.log`. Se os scripts `OnNodeStart` ou `OnNodeConfigured` forem especificados na configuração do cluster, verifique duas vezes se o script é executado com êxito por meio de mensagens de log.

Quando um cluster é criado, o nó principal precisa esperar que os nós de computação se juntem ao cluster antes que ele possa se juntar ao cluster. Por esse motivo, se os nós de computação não conseguirem se juntar ao cluster, o nó principal também falhará. É possível seguir um desses conjuntos de procedimentos, dependendo do tipo de nós computacionais usados, para solucionar esse tipo de problema:

Nós de computação

- Logs aplicáveis:
 - `/var/log/cloud-init-output.log`
 - `/var/log/slurmd.log`
- Se um nó de computação for iniciado, primeiro verifique o `/var/log/cloud-init-output.log`, que deve conter os logs de configuração semelhantes ao log `/var/log/chef-client.log` do nó principal. A maioria dos erros que ocorrem durante a configuração deve ter mensagens de erro localizadas no log `/var/log/cloud-init-output.log`. Se scripts de pré-instalação ou pós-instalação forem especificados na configuração do cluster, verifique se eles foram executados com êxito.
- Se você estiver usando uma AMI personalizada com modificação na configuração Slurm, talvez haja um erro relacionado ao Slurm que impeça o nó de computação de se juntar ao cluster. Para erros relacionados ao programador, verifique o log `/var/log/slurmd.log`.

Nós de computação dinâmicos:

- Pesquise no log ResumeProgram (`/var/log/parallelcluster/slurm_resume.log`) o nome do seu nó de computação para ver se alguma vez o ResumeProgram foi chamado com o nó. (Se o ResumeProgram nunca foi chamado, você pode verificar o log `slurmctld` (`/var/log/slurmctld.log`) para determinar se o Slurm já tentou chamar ResumeProgram com o nó).
- Observe que permissões incorretas em ResumeProgram podem fazer com que ResumeProgram falhe silenciosamente. Se você estiver usando uma AMI personalizada com modificações na configuração ResumeProgram, verifique se a ResumeProgram é de propriedade do usuário `slurm` e tem a permissão de `744 (rwxr--r--)`.
- Se ResumeProgram for chamado, verifique se uma instância foi executada para o nó. Se nenhuma instância foi iniciada, você pode ver uma mensagem de erro que descreve a falha na inicialização.
- Se a instância for executada, é possível que um problema tenha ocorrido durante o processo de configuração. Você deve ver o endereço IP privado e o ID da instância correspondentes no log ResumeProgram. Além disso, você pode ver os logs de configuração correspondentes para a instância específica. Para ter mais informações sobre como solucionar um erro de configuração com um nó de computação, consulte a próxima seção.

Nós de computação estáticos:

- Verifique o log `clustermgtd (/var/log/parallelcluster/clustermgtd)` para ver se foram iniciadas instâncias para o nó. Se elas não foram iniciadas, deve haver uma mensagem de erro clara detalhando a falha na inicialização.
- Se a instância for iniciada, haverá algum problema durante o processo de configuração. Você deve ver o endereço IP privado e o ID da instância correspondentes no log `ResumeProgram`. Além disso, você pode ver os logs de configuração correspondentes para a instância específica.

Nós de computação apoiados por instâncias spot:

- Se for a primeira vez que você usa instâncias spot e o trabalho permanece em um PD (estado pendente), verifique novamente o arquivo `/var/log/parallelcluster/slurm_resume.log`. É provável que você encontre um erro semelhante ao seguinte:

```
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 EC2 Spot Instances.
```

Ao usar instâncias Spot, uma função vinculada ao serviço `AWSServiceRoleForEC2Spot` deve existir na sua conta. Para criar essa função na sua conta usando a AWS CLI, execute o seguinte comando:

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

Para obter mais informações, consulte [Trabalho com Instâncias spot](#) no Guia do usuário AWS ParallelCluster e [Função vinculada ao serviço para solicitações de instâncias spot](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Solução de problemas inesperados de substituições e encerramentos de nós

Esta seção continua explorando como você pode solucionar problemas relacionados ao nó, especificamente quando um nó é substituído ou encerrado inesperadamente.

- Logs aplicáveis:
 - `/var/log/parallelcluster/clustermgtd` (nó principal)

- `/var/log/slurmctld.log` (nó principal)
- `/var/log/parallelcluster/computemgtd` (nó de computação)

Nós substituídos/encerrados inesperadamente

- Verifique no log `clustermgtd` (`/var/log/parallelcluster/clustermgtd`) se um `clustermgtd` substituiu ou encerrou um nó. Observe que `clustermgtd` trata de todas as ações normais de manutenção do nó.
- Se o `clustermgtd` substituiu ou encerrou um nó, deverá haver uma mensagem detalhando por que essa ação foi tomada no nó. Se o motivo estiver relacionado com o programador (por exemplo, o nó estiver em `DOWN`), verifique o log `slurmctld` para mais informações. Se o motivo for relacionado ao Amazon EC2, deve haver uma mensagem informativa detalhando o problema relacionado ao Amazon EC2 que exigiu a substituição.
- Se o `clustermgtd` não encerrou o nó, primeiro verifique se esse era um encerramento esperado pelo Amazon EC2, mais especificamente um encerramento pontual. O `computemgtd`, executado em um nó de computação, também pode encerrar um nó se `clustermgtd` for determinado como não saudável. Verifique no log `computemgtd` (`/var/log/parallelcluster/computemgtd`) se um `computemgtd` encerrou o nó.

Falha nos nós

- Verifique no log `slurmctld` (`/var/log/slurmctld.log`) para ver por que um trabalho ou um nó falhou. Observe que os trabalhos são automaticamente enfileirados novamente se um nó falhar.
- Se `slurm_resume` relatar que o nó foi iniciado e `clustermgtd` relatar após alguns minutos que não há nenhuma instância correspondente no Amazon EC2 para esse nó, o nó pode falhar durante a configuração. Para recuperar o log de um computador (`/var/log/cloud-init-output.log`), execute as seguintes etapas:
 - Envie um trabalho para permitir que o Slurm gere um novo nó.
 - Depois que o nó for iniciado, ative a proteção contra encerramento usando esse comando.

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --disable-api-termination
```

- Recupere a saída do console do nó com esse comando.

```
$ aws ec2 get-console-output --instance-id i-1234567890abcdef0 --output text
```


Substituindo, encerrando ou desligando instâncias e nós problemáticos

- Logs aplicáveis:
 - `/var/log/parallelcluster/clustermgtd` (nó principal)
 - `/var/log/parallelcluster/slurm_suspend.log` (nó principal)
- Na maioria dos casos, `clustermgtd` processa todas as ações esperadas de encerramento da instância. Examine o log `clustermgtd` para ver por que ele não conseguiu substituir ou encerrar um nó.
- Se os nós dinâmicos falharem no [Propriedades do SlurmSettings](#), consulte o log `SuspendProgram` para ver se o `SuspendProgram` foi chamado pelo `slurmctld` com o nó específico como argumento. Observe que o `SuspendProgram` não executa nenhuma ação. Em vez disso, ele só cria logs quando é chamado. Todo encerramento da instância e restauração de `NodeAddr` são feitos por `clustermgtd`. O Slurm coloca os nós de volta em um estado `POWER_SAVING` depois do `SuspendTimeout` automaticamente.
- Se os nós de computação falharem continuamente devido a falhas de bootstrap, verifique se eles estão sendo iniciados com a opção [Modo protegido por cluster do Slurm](#) habilitada. Se o modo protegido não estiver ativado, modifique as configurações do modo protegido para ativar o modo protegido. Solucione problemas e corrija o script de bootstrap.

Status **Inactive** da fila (partição)

Se você executar `sinfo` e a saída mostrar filas com status `AVAIL` de `inact`, seu cluster pode ter o [Modo protegido por cluster do Slurm](#) ativado e a fila foi definida para o estado `INACTIVE` por um período de tempo predefinido.

Solucionando outros problemas conhecidos de nós e tarefas

Outro tipo de problema conhecido é que AWS ParallelCluster pode falhar na alocação de trabalhos ou na tomada de decisões de escalabilidade. Com esse tipo de problema, AWS ParallelCluster apenas inicia, encerra ou mantém recursos de acordo com as instruções do Slurm. Para esses problemas, verifique o log `slurmctld` para solucioná-los.

Grupos de posicionamento e problemas de execução de instâncias

Para obter a menor latência entre os nós, use um grupo de posicionamento. Um grupo de posicionamento garante que suas instâncias estejam no mesmo suporte principal de rede. Se não houver instâncias suficientes disponíveis quando a solicitação é feita, um erro

`InsufficientInstanceCapacity` será gerado. Para reduzir a possibilidade de receber um erro ao usar grupos de posicionamento de cluster, defina o parâmetro [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#) / [Enabled](#) para `false`.

Para obter controle adicional sobre o acesso à capacidade, considere [lançar instâncias com ODCR \(On-Demand Capacity Reservations\)](#).

Para obter mais informações, consulte [Solução de problemas de execução de instâncias](#) e [Funções e limitações de placement groups](#) no Guia do usuário do Amazon EC2 par instâncias do Linux.

Diretórios que não podem ser substituídos

Os diretórios a seguir são compartilhados entre os nós e não podem ser substituídos.

- `/home`- Isso inclui a pasta inicial padrão do usuário (`/home/ec2_user` no Amazon Linux RedHat, `/home/centos` no CentOS e `/home/ubuntu` no Ubuntu).
- `/opt/intel` - Isso inclui Intel MPI, Intel Parallel Studio e arquivos relacionados.
- `/opt/slurm` - Isso inclui o Slurm Workload Manager e arquivos relacionados. (Condicional, somente se `Scheduler: slurm`.)

Solucionar problemas no NICE DCV

Tópicos

- [Logs para NICE DCV](#)
- [Problemas com o Ubuntu NICE DCV](#)

Logs para NICE DCV

Os logs do NICE DCV são gravados em arquivos no diretório `/var/log/dcv/`. A revisão desses logs pode ajudar a solucionar problemas.

O tipo de instância deve ter pelo menos 1,7 gibibytes (GiB) de RAM para executar o NICE DCV. Os tipos de instância nano e micro não têm memória suficiente para executar o NICE DCV.

AWS ParallelCluster cria fluxos de log NICE DCV em grupos de log. Você pode visualizar esses registros no CloudWatch console Painéis personalizados ou grupos de registros. Para obter mais informações, consulte [Painel do Amazon CloudWatch](#) e [Integração com Amazon CloudWatch Logs](#).

Problemas com o Ubuntu NICE DCV

Ao executar o Gnome Terminal em uma sessão NICE DCV no Ubuntu, você pode não ter acesso automático ao ambiente do usuário disponibilizado pelo AWS ParallelCluster por meio do shell de login. O ambiente do usuário fornece módulos de ambiente, como openmpi ou intelmpi, e outras configurações do usuário.

As configurações padrão do Gnome Terminal evitam que o shell inicie como um shell de login. Isso significa que os perfis de shell não são fornecidos automaticamente e o ambiente do usuário AWS ParallelCluster não é carregado.

Para obter o perfil do shell corretamente e acessar o ambiente do usuário AWS ParallelCluster, realize um dos seguintes procedimentos:

- Altere as configurações do terminal padrão:
 1. Escolha o menu Editar no terminal Gnome.
 2. Selecione Preferências, e em seguida, Perfis.
 3. Escolha Comando e selecione Executar comando como shell de login.
 4. Abrir um novo terminal.
- Use a linha de comando para obter os perfis disponíveis:

```
$ source /etc/profile && source $HOME/.bashrc
```

Solução de problemas em clusters com integração AWS Batch

Esta seção é relevante para clusters com integração com programadores AWS Batch.

Tópicos

- [Problemas no nó principal](#)
- [Problemas de computação](#)
- [Falhas de trabalhos](#)
- [Erro de tempo limite de conexão no URL do endpoint](#)

Problemas no nó principal

Você pode solucionar problemas de configuração do nó principal da mesma forma que um cluster Slurm (exceto para logs Slurm específicos). Para obter mais informações sobre esses problemas, consulte [Nó principal](#).

Problemas de computação

AWS Batch gerencia os aspectos de escalabilidade e computação de seus serviços. Se você encontrar problemas relacionados à computação, consulte a documentação de AWS Batch [solução de problemas](#) para obter ajuda.

Falhas de trabalhos

Se um trabalho falhar, você poderá executar o comando [awsbout](#) para recuperar a saída do trabalho. Você também pode executar o [awsbstat](#) comando para obter um link para os registros de trabalhos armazenados pela Amazon CloudWatch.

Erro de tempo limite de conexão no URL do endpoint

Se trabalhos paralelos de vários nós falharem com um erro: `Connect timeout on endpoint URL`:

- No log `awsbout` de saída, verifique se o trabalho tem vários nós paralelos à saída: `Detected 3/3 compute nodes. Waiting for all compute nodes to start.`
- Verifique se a sub-rede dos nós de computação é pública.

Os trabalhos paralelos de vários nós não suportam o uso de sub-redes públicas ao usarem AWS Batch e AWS ParallelCluster. Use uma sub-rede privada para seus nós e trabalhos de computação. Para obter mais informações, consulte [Considerações sobre o ambiente de computação](#) no Guia do usuário do AWS Batch. Para configurar uma sub-rede privada para seus nós de computação, consulte [AWS ParallelCluster com programador AWS Batch](#).

Solução de problemas de integração de vários usuários com o Active Directory

Esta seção é relevante para clusters integrados a um Active Directory.

Se o recurso de integração do Active Directory não estiver funcionando conforme o esperado, os registros do SSSD podem fornecer informações úteis de diagnóstico. Esses logs estão localizados em `/var/log/sss` nos nós do cluster. Por padrão, eles também são armazenados no grupo de CloudWatch logs da Amazon de um cluster.

Tópicos

- [Solução de problemas específicos do Active Directory](#)
- [Habilitar modo de depuração](#)
- [Como passar do LDAPS para o LDAP](#)
- [Como desabilitar a verificação do certificado do servidor LDAPS](#)
- [Como fazer login com uma chave SSH em vez de uma senha](#)
- [Como redefinir uma senha de usuário e senhas expiradas](#)
- [Como verificar o domínio associado](#)
- [Como solucionar problemas com certificados](#)
- [Como verificar se a integração com o Active Directory está funcionando](#)
- [Como solucionar problemas de login em nós de computação](#)
- [Problemas conhecidos com trabalhos SimCenter StarCCM+ em um ambiente multiusuário](#)
- [Problemas conhecidos com a resolução do nome de usuário](#)
- [Como resolver problemas de criação do diretório inicial](#)

Solução de problemas específicos do Active Directory

Esta seção é relevante para a solução de problemas específicos de um tipo de Active Directory.

Simple AD

- O valor `DomainReadOnlyUser` deve corresponder à pesquisa básica do diretório Simple AD para usuários:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

Anote `cn` para `Users`.

- O usuário administrador padrão é `Administrator`.
- `Ldapsearch` requer o nome NetBIOS antes do nome de usuário.

A sintaxe Ldapsearch deve ser a seguinte:

```
$ 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

- O valor `DomainReadOnlyUser` deve corresponder à pesquisa básica do diretório AWS Managed Microsoft AD para usuários:

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

- O usuário administrador padrão é Admin.
- A sintaxe Ldapsearch deve ser a seguinte:

```
$ ldapsearch -x -D "Admin" -w "Password" -H ldap://192.0.2.103 \  
-b "ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

Habilitar modo de depuração

Os registros de depuração do SSSD podem ser úteis para solucionar problemas. Para ativar o modo de depuração, você deve atualizar o cluster com as seguintes alterações feitas na configuração do cluster:

```
DirectoryService:  
  AdditionalSssdConfigs:  
    debug_level: "0x1ff"
```

Como passar do LDAPS para o LDAP

A mudança do LDAPS (LDAP com TLS/SSL) para o LDAP é desencorajada porque o LDAP sozinho não fornece nenhuma criptografia. No entanto, ele pode ser útil para fins de teste e solução de problemas.

Você pode restaurar a configuração anterior do cluster atualizando o cluster com a definição de configuração anterior.

Para migrar do LDAPS para o LDAP, você deve atualizar o cluster com as seguintes alterações na configuração do cluster:

```
DirectoryService:
  LdapTlsReqCert: never
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

Como desabilitar a verificação do certificado do servidor LDAPS

Pode ser útil desativar temporariamente a verificação do certificado do servidor LDAPS no nó principal, para fins de teste ou solução de problemas.

Você pode restaurar a configuração anterior do cluster atualizando o cluster com a definição de configuração anterior.

Para desativar a verificação do certificado do servidor LDAPS, você deve atualizar o cluster com as seguintes alterações na configuração do cluster:

```
DirectoryService:
  LdapTlsReqCert: never
```

Como fazer login com uma chave SSH em vez de uma senha

A chave SSH é criada no `/home/$user/.ssh/id_rsa` após a primeira vez que você efetua login com uma senha. Para fazer login com a chave SSH, você deve fazer login com sua senha, copiar a chave SSH localmente e usá-la para SSH sem senha, como de costume:

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY $username@$head_node_ip
```

Como redefinir uma senha de usuário e senhas expiradas

Se um usuário perder o acesso a um cluster, sua [senha AWS Managed Microsoft AD pode ter expirado](#).

Para redefinir a senha, execute o seguinte comando com um usuário e uma função que tenham permissão de gravação no diretório:

```
$ aws ds reset-user-password \
```

```
--directory-id "d-abcdef01234567890" \  
--user-name "USER_NAME" \  
--new-password "NEW_PASSWORD" \  
--region "region-id"
```

Se você redefinir a senha para o [DirectoryService](#) / [DomainReadOnlyUser](#):

1. Certifique-se de atualizar o segredo [DirectoryService](#) / [PasswordSecretArn](#) com a nova senha.
2. Atualize o cluster para obter o novo valor secreto:
 - a. Pare a frota de computadores com o comando `pcluster update-compute-fleet`.
 - b. Execute o comando a seguir de dentro do nó principal do cluster.

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

Após a redefinição da senha e a atualização do cluster, o acesso do usuário ao cluster deve ser restaurado.

Para obter mais informações, consulte [Redefinir uma senha de usuário](#) no Guia de administração do AWS Directory Service.

Como verificar o domínio associado

O comando a seguir deve ser executado em uma instância associada ao domínio, não no nó principal.

```
$ 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
```

Como solucionar problemas com certificados

Quando a comunicação LDAPS não está funcionando, pode ser devido a erros na comunicação TLS, o que, por sua vez, pode ser devido a problemas com certificados.

Notas sobre certificados:

- O certificado especificado na configuração do cluster `LdapTlsCaCert` deve ser um pacote de certificados PEM contendo os certificados de toda a cadeia de certificados de autoridade (CA) que emitiu certificados para os controladores de domínio.
- Um pacote de certificados PEM é um arquivo feito da concatenação de certificados PEM.
- Um certificado no formato PEM (normalmente usado no Linux) é equivalente a um certificado no formato base64 DER (normalmente exportado pelo Windows).
- Se o certificado para controladores de domínio for emitido por uma CA subordinada, o pacote de certificados deverá conter o certificado da CA subordinada e raiz.

Etapas de verificação de problemas:

As etapas de verificação a seguir pressupõem que os comandos sejam executados de dentro do nó principal do cluster e que o controlador de domínio esteja acessível em `SERVER:PORT`.

Para solucionar um problema relacionado aos certificados, siga estas etapas de verificação:

Etapas de verificação:

1. Verifique a conexão com os controladores de domínio do Active Directory:

Verifique se você pode se conectar a um controlador de domínio. Se essa etapa for bem-sucedida, a conexão SSL com o controlador de domínio será bem-sucedida e o certificado será verificado. Seu problema não está relacionado aos certificados.

Se essa etapa falhar, prossiga com a próxima verificação.

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

2. Revise a verificação do certificado:

Verifique se o pacote de certificados CA local pode validar o certificado fornecido pelo controlador de domínio. Se essa etapa for bem-sucedida, seu problema não está relacionado aos certificados, mas a outros problemas de rede.

Se essa etapa falhar, prossiga com a próxima verificação.

```
$ openssl verify -verbose -  
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

3. Verifique o certificado fornecido pelos controladores de domínio do Active Directory:

Verifique se o conteúdo do certificado fornecido pelos controladores de domínio é o esperado. Se essa etapa for bem-sucedida, você provavelmente terá problemas com o certificado CA usado para verificar os controladores. Vá para a próxima etapa de solução de problemas.

Se essa etapa falhar, você deverá corrigir o certificado emitido para os controladores de domínio e reexecutar as etapas de solução de problemas.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

4. Verifique o conteúdo de um certificado:

Verifique se o conteúdo do certificado fornecido pelos controladores de domínio é o esperado. Se essa etapa for bem-sucedida, você provavelmente terá problemas com o certificado CA usado para verificar os controladores. Vá para a próxima etapa de solução de problemas.

Se essa etapa falhar, você deverá corrigir o certificado emitido para os controladores de domínio e reexecutar as etapas de solução de problemas.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

5. Verifique o conteúdo do pacote de certificados CA local:

Verifique se o conteúdo do pacote de certificados CA local usado para validar o certificado dos controladores de domínio está conforme o esperado. Se essa etapa for bem-sucedida, você provavelmente terá problemas com o certificado fornecido pelos controladores de domínio.

Se essa etapa falhar, você deverá corrigir o pacote de certificado de CA emitido para os controladores de domínio e reexecutar as etapas de solução de problemas.

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

Como verificar se a integração com o Active Directory está funcionando

Se as duas verificações a seguir forem bem-sucedidas, a integração com o Active Directory está funcionando.

Verificações:

1. Você pode descobrir usuários definidos no diretório:

De dentro do nó principal do cluster, como um `ec2-user`:

```
$ getent passwd $ANY_AD_USER
```

2. Você pode usar SSH no nó principal fornecendo a senha do usuário:

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

Se a verificação um falhar, esperamos que a verificação dois também falhe.

Verificações adicionais para solução de problemas:

- Verifique se o usuário existe no diretório.
- Habilitar [debug logging](#).
- Considere desativar temporariamente a criptografia [migrando do LDAPS para o LDAP](#) para descartar problemas de LDAPS.

Como solucionar problemas de login em nós de computação

Esta seção é relevante para fazer login em nós de computação em clusters integrados ao Active Directory.

Com AWS ParallelCluster, os logins com senha nos nós de computação do cluster são desativados por design.

Todos os usuários devem usar sua própria chave SSH para fazer login nos nós de computação.

Os usuários podem recuperar sua chave SSH no nó principal após a primeira autenticação (por exemplo, login), se [GenerateSshKeysForUsers](#) estiver habilitadas na configuração do cluster.

Quando os usuários se autenticam no nó principal pela primeira vez, eles podem recuperar chaves SSH que são geradas automaticamente para eles como usuários do diretório. Também são criados diretórios pessoais para o usuário. Isso também pode acontecer na primeira vez que um sudo-usuário muda para um usuário no nó principal.

Se um usuário não estiver conectado ao nó principal, as chaves SSH não serão geradas e o usuário não poderá fazer login nos nós de computação.

Problemas conhecidos com trabalhos SimCenter StarCCM+ em um ambiente multiusuário

Esta seção é relevante para trabalhos lançados em um ambiente multiusuário pelo software de dinâmica de fluidos computacional Simcenter StarCCM+ da Siemens.

Se você executar trabalhos StarCCM+ v16 configurados para usar o IntelMPI incorporado, por padrão, os processos MPI serão inicializados usando SSH.

Devido a um [bug do Slurm](#) que faz com que a resolução do nome de usuário esteja errada, os trabalhos podem falhar com um erro como `error setting up the bootstrap proxies`. Esse bug afeta somente AWS ParallelCluster versões 3.1.1 e 3.1.2.

Para evitar que isso ocorra, force o IntelMPI a usar o Slurm como método de bootstrap do MPI. Exporte a variável de ambiente `I_MPI_HYDRA_BOOTSTRAP=slurm` para o script de trabalho que inicia o StarCCM+, conforme descrito na [documentação oficial do IntelMPI](#).

Problemas conhecidos com a resolução do nome de usuário

Esta seção é relevante para recuperar nomes de usuário em trabalhos.

Devido a um [bug conhecido no Slurm](#), o nome de usuário recuperado em um processo de trabalho pode ser `nobody` se você executar um trabalho sem `srun`. Esse bug afeta somente AWS ParallelCluster versões 3.1.1 e 3.1.2.

Por exemplo, se você executar o comando `sbatch --wrap 'srun id'` como usuário do diretório, o nome de usuário correto será retornado. No entanto, se você executar o `sbatch --wrap 'id'` como usuário do diretório, `nobody` poderá ser retornado como nome de usuário.

É possível usar as seguintes soluções alternativas.

1. Inicie seu trabalho com 'srun' em vez de 'sbatch', se possível.
2. Ative a enumeração SSSD definindo a configuração [AdditionalSssdConfigs](#) no cluster da seguinte forma.

```
AdditionalSssdConfigs:  
  enumerate: true
```

Como resolver problemas de criação do diretório inicial

Esta seção é relevante para problemas de criação de diretórios iniciais.

Se você ver erros como o mostrado no exemplo a seguir, um diretório inicial não foi criado para você quando você fez login pela primeira vez no nó principal. Ou, um diretório inicial não foi criado para você quando você mudou pela primeira vez de um sudoer para um usuário do Active Directory no nó principal.

```
$ 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
```

A falha na criação do diretório inicial pode ser causada pelos pacotes oddjob e oddjob-mkhomedir instalados no nó principal do cluster.

Sem um diretório inicial e uma chave SSH, o usuário não pode enviar trabalhos ou SSH para os nós do cluster.

Se você precisar dos pacotes oddjob em seu sistema, verifique se o serviço oddjobd está em execução e atualize os arquivos de configuração do PAM para garantir que o diretório inicial seja criado. Para fazer isso, execute os comandos no nó principal, conforme mostrado no exemplo a seguir.

```
sudo systemctl start oddjobd  
sudo authconfig --enablemkhomedir --updateall
```

Se você não precisar dos pacotes `oddjob` em seu sistema, desinstale-os e atualize os arquivos de configuração do PAM para garantir que o diretório inicial seja criado. Para fazer isso, execute os comandos no nó principal, conforme mostrado no exemplo a seguir.

```
sudo yum remove -y oddjob oddjob-mkhomedir
sudo authconfig --enablemkhomedir --updateall
```

Solução de problemas de AMI personalizada

Ao usar uma AMI personalizada, você pode ver os seguintes avisos:

```
"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 tiver certeza de que a AMI correta está sendo usada, você pode ignorar esses avisos.

Se você não quiser ver esses avisos no futuro, marque a AMI personalizada com as seguintes tags, onde *my-os* é uma das `alinux2`, `ubuntu2204`, `ubuntu2004`, `centos7`, ou `rhel8` e `"3.7.0"` é a versão `pcluster` em uso:

```
$ aws ec2 create-tags \
  --resources ami-yourcustomAmi \
  --tags Key="parallelcluster:version",Value="3.7.0"
Key="parallelcluster:os",Value="my-os"
```

Solução de problemas de tempo limite de atualização de cluster quando **cfn-hup** não está em execução

O auxiliar `cfn-hup` é um daemon que detecta alterações em metadados de recursos e executa ações especificadas pelo usuário quando uma alteração é detectada. É assim que você faz as atualizações de configuração em suas instâncias do Amazon EC2 em execução pela ação do API `UpdateStack`.

Atualmente, o daemon `cfn-hup` é lançado pelo `supervisord`. Mas após o lançamento, o processo `cfn-hup` é separado do controle `supervisord`. Se o daemon `cfn-hup` for encerrado por uma causa externa, ele não será reiniciado automaticamente. Se `cfn-hup` não estiver em execução, durante uma atualização do cluster, a CloudFormation pilha inicia o processo de atualização conforme o esperado, mas o procedimento de atualização não é ativado no nó principal e a pilha acaba atingindo o tempo limite. Nos logs do cluster `/var/log/chef-client`, você pode ver que a fórmula de atualização nunca é invocada.

Verifique e reinicie **cfn-hup** em caso de falhas

1. No nó principal, verifique se `cfn-hup` está em execução:

```
$ ps aux | grep cfn-hup
```

2. Verifique o log `cfn-hup` `/var/log/cfn-hup.log` e `/var/log/supervisord.log` no nó principal.

3. Se `cfn-hup` não estiver em execução, tente reiniciá-lo executando:

```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl  
start cfn-hup
```

Solução de problemas de rede

Problemas de cluster em uma única sub-rede pública

Verifique o `cloud-init-output.log` a partir de um dos nós de computação. Se você encontrar algo como o seguinte que indica que o nó está preso na inicialização Slurm, provavelmente é devido à falta de um endpoint da VPC do DynamoDB. Adicione o endpoint do DynamoDB. Para obter mais informações, consulte [AWS ParallelCluster em uma sub-rede única sem acesso à 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)
```

Falha na atualização do cluster na ação personalizada **onNodeUpdated**

Quando um script [HeadNode](#) / [CustomActions](#) / [OnNodeUpdated](#) falha, a atualização falha e o script não é executado no momento da reversão. É sua responsabilidade realizar manualmente as limpezas necessárias após a conclusão da reversão. Por exemplo, se o script `OnNodeUpdated` alterar o status de um campo em um arquivo de configuração (por exemplo, de `true` para `false`) e depois falhar, você precisará restaurar manualmente o valor do campo para o estado anterior à atualização (por exemplo, `false` para `true`). Para obter mais informações, consulte [Ações de bootstrap personalizadas](#).

Vendo erros com a configuração personalizada Slurm

A partir do AWS ParallelCluster versão 3.6.0, você não pode mais direcionar scripts individuais `prolog` ou `epilog` incluindo-os em uma configuração personalizada Slurm. No AWS ParallelCluster versão 3.6.0 e versões posteriores, você deve localizar scripts personalizados `prolog` e `epilog` e nas respectivas pastas `Prolog` e `Epilog`. Essas pastas são configuradas por padrão para apontar para:

- `Prolog` aponta para `/opt/slurm/etc/scripts/prolog.d/`.
- `Epilog` aponta para `/opt/slurm/etc/scripts/epilog.d/`.

Recomendamos que você mantenha o script do `prolog` `90_plcluster_health_check_manager` e o script do `epilog` `90_pcluster_noop` à disposição.

Slurm executa os scripts em ordem alfabética inversa. Tanto a pasta `Prolog` quanto a `Epilog` devem ter pelo menos um arquivo. Para obter mais informações, consulte [Personalização de configuração do Slurm](#) e [Slurm prolog e epilog](#).

Alarmes de cluster

O monitoramento da integridade do cluster é essencial para garantir o desempenho ideal. AWS ParallelCluster permite monitorar vários alarmes CloudWatch baseados no nó principal do cluster.

Esta seção fornece detalhes para cada tipo de alarme do cluster Head Node, incluindo suas convenções de nomenclatura, condições específicas que acionam alarmes e etapas de solução de problemas sugeridas.

A convenção de nomenclatura para alarmes de cluster é `CLUSTER_NAME-COMPONENT-METRIC`, por exemplo. `mycluster-HeadNode-Cpu`

- `CLUSTER_NAME-HeadNode`: sinaliza o status geral do nó principal. É vermelho se pelo menos um dos alarmes abaixo estiver.
- `CLUSTER_NAME-HeadNode-Health`: vermelho se houver pelo menos uma falha no EC2 Health Check. Em caso de alarme, sugerimos que você dê uma olhada em [Solucionar problemas de instâncias com falhas nas verificações de status](#).
- `CLUSTER_NAME-HeadNode-Cpu`: vermelho se a utilização da CPU for maior que 90%. Em caso de alarme, verifique os processos que estão consumindo mais a CPU `comps -aux --sort=%cpu | head -n 10`.
- `CLUSTER_NAME-HeadNode-Mem`: vermelho se a utilização da memória for maior que 90%. Em caso de alarme, verifique os processos que estão consumindo mais memória `comps -aux --sort=-%mem | head -n 10`.
- `CLUSTER_NAME-HeadNode-Disk`: vermelho se o espaço em disco ocupado for maior que 90% no caminho `/`. Em caso de alarme, verifique as pastas que consomem a maior parte do espaço com `du -h --max-depth=2 / 2> /dev/null | sort -hr`.

Suporte adicional

Para obter uma lista de problemas conhecidos, consulte a página principal [GitHub do Wiki](#) ou a página de [problemas](#).

Para problemas mais urgentes, entre em contato AWS Support ou abra um [novo GitHub problema](#).

AWS ParallelCluster política de suporte

AWS ParallelCluster suporta vários lançamentos ao mesmo tempo. Cada AWS ParallelCluster lançamento tem uma data programada de End of Support Life (EOSL). Após a data do EOSL, nenhum suporte ou manutenção adicional é fornecido para essa versão.

AWS ParallelCluster usa um esquema de `major.minor.patch` versão. Novos recursos, melhorias de desempenho, atualizações de segurança e correções de erros estão incluídos nas novas versões secundárias da versão principal mais recente. As versões secundárias são compatíveis com versões anteriores em uma versão principal. Para problemas críticos, AWS fornece correções por meio de lançamentos de patches, mas somente para as versões secundárias mais recentes de versões que não chegaram ao EOSL. Se você quiser usar as atualizações de uma nova versão, precisará atualizar para a nova versão secundária ou patch.

AWS ParallelCluster versões	Data de fim da vida útil suportada (EOSL)
3.0. <i>x</i>	31/03/2023
3.1. <i>x</i>	31/08/2023
3.2. <i>x</i>	31/01/2024
3.3. <i>x</i>	31/05/2024
3.4. <i>x</i>	28/06/2024
3.5. <i>x</i>	31/08/2024
3.6. <i>x</i>	30/11/2024
3.7. <i>x</i>	28/02/2025
3.8. <i>x</i>	30/06/2025
3.9. <i>x</i>	09/05/2025

Segurança em AWS ParallelCluster

A segurança para com a nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao AWS ParallelCluster, consulte [Serviços da AWS em escopo por programa de conformidade](#)
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço ou serviços específicos da AWS que você usa. Você também é responsável por diversos outros fatores relacionados, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação descreve como aplicar o modelo de responsabilidade compartilhada ao usar o AWS ParallelCluster. Os tópicos a seguir mostram como configurar o AWS ParallelCluster para atender aos seus objetivos de segurança e conformidade. Saiba também como usar a AWS ParallelCluster de uma maneira que ajude você a monitorar e proteger os recursos do AWS.

Tópicos

- [Informações de segurança para serviços usados pela AWS ParallelCluster](#)
- [Proteção de dados no AWS ParallelCluster](#)
- [Identity and Access Management para o AWS ParallelCluster](#)
- [Validação de conformidade do AWS ParallelCluster](#)
- [Impor uma versão mínima do TLS 1.2](#)

Informações de segurança para serviços usados pela AWS ParallelCluster

- [Segurança no Amazon EC2](#)
- [Segurança no Amazon API Gateway](#)
- [Segurança em AWS Batch](#)
- [Segurança em AWS CloudFormation](#)
- [Segurança no Amazon CloudWatch](#)
- [Segurança em AWS CodeBuild](#)
- [Segurança no Amazon DynamoDB](#)
- [Segurança no Amazon ECR](#)
- [Segurança no Amazon ECS](#)
- [Segurança no Amazon EFS](#)
- [Segurança no FSx for Lustre](#)
- [Segurança na AWS Identity and Access Management \(IAM\)](#)
- [Segurança no EC2 Image Builder](#)
- [Segurança em AWS Lambda](#)
- [Segurança no Amazon Route 53](#)
- [Segurança no Amazon SNS](#)
- [Segurança no Amazon SQS \(para a AWS ParallelCluster versão 2.x.\)](#)
- [Segurança no Amazon S3](#)
- [Segurança no Amazon VPC](#)

Proteção de dados no AWS ParallelCluster

O [modelo de responsabilidade compartilhada](#) da AWS se aplica à proteção de dados no AWS ParallelCluster. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que você usa. Para ter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para ter

mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and GDPR](#) no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão dos Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui trabalhar com a AWS ParallelCluster ou outros Serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia de dados

Um recurso fundamental de qualquer serviço seguro é que as informações sejam criptografadas quando não estão sendo usadas ativamente.

Criptografia em repouso

A própria AWS ParallelCluster não armazena nenhum dado do cliente além das credenciais de que precisa para interagir com os serviços da AWS em nome do usuário.

Para dados nos nós do cluster, os dados podem ser criptografados em repouso.

Para volumes do Amazon EBS, a criptografia é configurada usando as configurações [EbsSettings/Encrypted](#) e [EbsSettings/KmsKeyId](#) na seção [EbsSettings](#). Para ter mais informações, consulte [Criptografia do Amazon EBS](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Para volumes do Amazon EFS, a criptografia é configurada usando as configurações [EfsSettings/Encrypted](#) e [EfsSettings/KmsKeyId](#) na seção [EfsSettings](#). Para obter mais informações, consulte [Como funciona a Criptografia de dados em repouso](#) no Guia do usuário do Amazon Elastic File System.

Para sistemas de arquivos FSx for Lustre, a criptografia de dados em repouso é ativada automaticamente ao criar um sistema de arquivos Amazon FSx. Para obter mais informações, consulte [Criptografia de dados em repouso](#) no Guia do usuário do Amazon FSx para Lustre.

Para os tipos de instância com volumes NVMe, os dados nos volumes de armazenamento de instância de NVMe são criptografados usando uma criptografia XTS-AES-256 implementada em um módulo de hardware na instância. As chaves de criptografia são geradas usando o módulo de hardware e são exclusivas para cada dispositivo de armazenamento de instâncias de NVMe. Todas as chaves de criptografia são destruídas quando a instância é interrompida ou encerrada e não podem ser recuperadas. Você não pode desativar essa criptografia e não pode fornecer sua própria chave de criptografia. Para obter mais informações, consulte [Criptografia por padrão](#) no Manual do usuário do Amazon EC2 para instâncias do Linux.

Se você usar a AWS ParallelCluster para invocar um serviço da AWS que transmita dados do cliente ao computador local para armazenamento, consulte o capítulo Segurança e conformidade no Guia do usuário desse serviço para obter informações sobre como esses dados são armazenados, protegidos e criptografados.

Criptografia em trânsito

Por padrão, todos os dados transmitidos do computador cliente que executa os endpoints de serviço da AWS ParallelCluster e da AWS são criptografados enviando tudo por meio de uma conexão

HTTPS/TLS. O tráfego entre os nós no cluster pode ser criptografado automaticamente, dependendo dos tipos de instância selecionados. Para mais informações, consulte [Criptografia em trânsito](#) no Manual do usuário do Amazon EC2 para instâncias do Linux.

Consulte também

- [Proteção de dados no Amazon EC2](#)
- [Proteção de dados no EC2 Image Builder](#)
- [Proteção de dados no AWS CloudFormation](#)
- [Proteção de dados no Amazon EFS](#)
- [Proteção de dados no Amazon S3](#)
- [Proteção de dados no FSx para Lustre](#)

Identity and Access Management para o AWS ParallelCluster

A AWS ParallelCluster usa funções para acessar seus recursos da AWS e os serviços dela. As políticas de instância e usuário usadas pela AWS ParallelCluster para conceder permissões estão documentadas em [AWS Identity and Access Management permissões em AWS ParallelCluster](#).

A única diferença importante é como você faz a autenticação ao usar um usuário do padrão e credenciais de longo prazo. Embora um usuário do AWS exija uma senha para acessar o console de um serviço da , esse mesmo usuário do exige um par de chaves de acesso para executar as mesmas operações usando a AWS ParallelCluster. Todas as outras credenciais de curto prazo são usadas da mesma maneira que são usadas com o console.

As credenciais usadas pela AWS ParallelCluster são armazenadas em arquivos de texto não criptografado e não são criptografadas.

- O arquivo `$HOME/.aws/credentials` armazena credenciais de longo prazo necessárias para acessar os recursos da AWS. Isso inclui o ID de chave de acesso e a chave de acesso secreta.
- As credenciais de curto prazo, como as de funções que você assume, ou que são para serviços do AWS IAM Identity Center, também são armazenadas nas pastas `$HOME/.aws/cli/cache` e `$HOME/.aws/sso/cache`, respectivamente.

Mitigação de riscos

- É altamente recomendável que você configure as permissões do sistema de arquivos na pasta \$HOME/.aws e suas respectivas pastas e arquivos filho para restringir o acesso somente a utilizadores autorizados.
- Use funções com credenciais temporárias sempre que possível para reduzir a oportunidade de danos se as credenciais estiverem comprometidas. Use credenciais de longo prazo apenas para solicitar e atualizar credenciais de função de curto prazo.

Validação de conformidade do AWS ParallelCluster

Audidores externos avaliam a segurança e a conformidade dos serviços da AWS como parte de vários programas de conformidade da AWS. Usar o AWS ParallelCluster para acessar um serviço não altera a conformidade desse serviço.

Para ver a lista de serviços da AWS no escopo de programas de compatibilidade específicos, consulte [Serviços da AWS no escopo por programa de compatibilidade](#). Para obter informações gerais, consulte [Programas de conformidade da AWS](#).

É possível baixar relatórios de auditoria externa usando o AWS Artifact. Para obter mais informações, consulte [Fazer download de relatórios no AWS Artifact](#).

Sua responsabilidade de conformidade ao usar o AWS ParallelCluster é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- [Guias de início rápido de segurança e compatibilidade](#): esses guias de implantação apresentam considerações de arquitetura e etapas para a implantação de ambientes básicos focados na segurança e na compatibilidade na AWS.
- [Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services AWS Whitepaper](#): este whitepaper descreve como as empresas podem utilizar a AWS para criar aplicações em conformidade com a HIPAA.
- [Recursos de compatibilidade da AWS](#): essa coleção de manuais e guias pode ser válida para seu setor e local.
- [Avaliar recursos com regras](#) no Guia do desenvolvedor do AWS Config: o serviço AWS Config avalia como as configurações de recursos estão em compatibilidade com práticas internas, diretrizes do setor e regulamentos.

- [AWS Security Hub](#): esse serviço da AWS fornece uma visão abrangente do estado de sua segurança na AWS que ajuda você a conferir sua conformidade com padrões e práticas recomendadas de segurança do setor.

Impor uma versão mínima do TLS 1.2

Para aumentar a segurança ao se comunicar com serviços da AWS, é necessário configurar a sua AWS ParallelCluster para usar o TLS 1.2 ou posterior. Quando você usa a AWS ParallelCluster, o Python é usado para definir a versão do TLS.

Para garantir que a AWS ParallelCluster não use uma versão do TLS anterior ao TLS 1.2, talvez seja necessário recompilar o OpenSSL para impor esse mínimo e recompilar o Python para usar o OpenSSL recém-criado.

Determinar os protocolos atualmente compatíveis

Primeiro, crie um certificado autoassinado a ser usado para o servidor de teste e o Python SDK usando OpenSSL.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Depois, crie um servidor de teste usando OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

Em uma nova janela do terminal, crie um ambiente virtual e instale o Python SDK.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Crie um script Python chamado `check.py` que usa a biblioteca HTTP subjacente do 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'))
```

Execute o novo script.

```
$ python check.py
```

Ele exibe detalhes sobre a conexão feita. Procure “Protocol :” na saída. Se a saída for “TLSv1.2” ou posterior, o SDK definirá como padrão TLS v1.2 ou posterior. Se for uma versão anterior, é necessário recompilar o OpenSSL e recompilar o Python.

No entanto, mesmo que a instalação do Python defina como padrão TLS v1.2 ou posterior, ainda será possível que o Python renegocie para uma versão anterior ao TLS v1.2 se o servidor não for compatível com TLS v1.2 ou posterior. Para verificar se o Python não renegocia automaticamente para versões anteriores, reinicie o servidor de teste com o seguinte.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Se estiver usando uma versão anterior do OpenSSL, talvez você não tenha o sinalizador `-no_tls_3` disponível. Se esse for o caso, remova o sinalizador porque a versão do OpenSSL que você está usando não é compatível com TLS v1.3. Execute novamente o script Python.

```
$ python check.py
```

Se a instalação do Python não renegocia para versões anteriores ao TLS 1.2 corretamente, você deve receber um erro SSL.

```
$ urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='localhost',
port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL:
UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

Se você conseguir fazer uma conexão, será necessário recompilar o OpenSSL e o Python para desabilitar a negociação de protocolos anteriores ao TLS v1.2.

Compilar OpenSSL e Python

Para garantir que a AWS ParallelCluster não negocie nada anterior ao TLS 1.2, é necessário recompilar o OpenSSL e o Python. Para fazer isso, copie o seguinte conteúdo para criar um script e executá-lo.

```
#!/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
```

Isso compila uma versão do Python que tem um OpenSSL vinculado estaticamente que não negocia automaticamente nada anterior ao TLS 1.2. Isso também instala o OpenSSL no diretório `/opt/openssl-with-min-tls1_2` e instala o Python no diretório `/opt/python-with-min-tls1_2`. Depois de executar esse script, verifique a instalação da nova versão do Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Isso deve imprimir o seguinte.

Python 3.8.1

Para verificar se essa nova versão do Python não negocia uma versão anterior ao TLS 1.2, execute novamente as etapas em [Determinar os protocolos atualmente compatíveis](#) usando a versão do Python recém-instalada (ou seja, `/opt/python-with-min-tls1_2/bin/python3`).

Notas de release e histórico de documentos

A tabela a seguir descreve as principais atualizações e novos atributos para o Guia do usuário do AWS ParallelCluster . Também atualizamos a documentação com frequência para abordar os comentários enviados por você.

Alteração	Descrição	Data
AWS ParallelCluster Lançada a versão 2024.04.0 da interface do usuário	<p>AWS ParallelCluster Lançada a versão 2024.04.0 da interface do usuário.</p> <p>Recursos:</p> <ul style="list-style-type: none">• Suporte adicionado para a AWS ParallelCluster versão 3.9.1• Registro de alterações completo	17 de abril de 2024
AWS ParallelCluster versão 3.9.1 lançada	<p>Temos o prazer de anunciar o lançamento da versão AWS ParallelCluster 3.9.1</p> <p>Para atualizar, insira o seguinte: <code>sudo pip install --upgrade aws-parallelcluster</code></p> <p>Correções de erros</p> <ul style="list-style-type: none">• Remova a exclusão recursiva do mountdir de armazenamento compartilhado ao desmontar sistemas de arquivos como	11 de abril de 2024

parte da operação de
atualização do cluster.

[AWS ParallelCluster versão 3.9.1 lançada](#)

Temos o prazer de anunciar
o lançamento da versão AWS
ParallelCluster 3.9.1

11 de abril de 2024

Para atualizar, insira o
seguinte: `sudo pip
install --upgrade
aws-parallelcluster`

Correções de erros

- Remova a exclusão
recursiva do mountdir de
armazenamento compartil
hado ao desmontar
sistemas de arquivos como
parte da operação de
atualização do cluster.

[AWS ParallelCluster Lançada a versão 2024.03.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2024.03.0 da interface do usuário.

12 de março de 2024

Recursos:

- Adicionado suporte para a AWS ParallelCluster versão 3.9.0
- Adicionado suporte para Ubuntu 22.04 e Red Hat Enterprise Linux 9
- Ubuntu 18.04 obsoleto

Correções de bugs

- Foi corrigido o problema que fazia com que alguns clusters não aparecessem ao usar muitos clusters

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster versão 3.9.0 lançada](#)

Temos o prazer de anunciar o lançamento da versão AWS ParallelCluster 3.9.0

5 de março de 2024

Para atualizar, insira o seguinte: `sudo pip install --upgrade aws-parallelcluster`

Melhorias:

- Adicione o parâmetro de configuração `DeploymentSettings/DefaultUserHome` para permitir que os usuários movam o diretório inicial do usuário padrão para `/local/home` em vez de `/home` (padrão).
- Permita a atualização `MinCount Queue` e a `ComputeResource` configuração dos parâmetros sem a necessidade de interromper a frota de computação. `MaxCount` Agora é possível atualizá-los configurando `Scheduling/SlurmSettings/QueueUpdateStrategy` para `TERMINAR`. AWS ParallelCluster encerrará somente os nós removidos durante um redimensionamento da capacidade

e do cluster realizado por meio de uma atualização do cluster.

- Permita atualizar o armazenamento externo compartilhado do tipo Efs, FsxLustre, FsxOntap, FsxOpenZfs e FileCache sem substituir a frota de computação e login.
- Adicione suporte para RHEL9.
- Adicione suporte para Rocky Linux 9 conforme CustomAmi criado por meio do build-image processo. Nenhuma AMI pública oficial do AWS ParallelCluster Rocky9 Linux está disponível no momento.
- Remover CommunicationParameters da lista de negação de Slurm configurações personalizadas.
- Adicione um DeploymentSettings/DisableSudoAccessForDefaultUser parâmetro para desativar o acesso sudo do usuário padrão em sistemas operacionais compatíveis.
- Alterações nos sistemas de arquivos FSx for Lustre

ParallelCluster criados por:
Altere a versão do servidor
Lustre para 2.15.

- Adicione a possibilidade de escolher entre drivers Nvidia de código aberto e fechado ao criar uma AMI, por meio do atributo `['cluster']['nvidia']['kernel_open']` `cookbook` `node`.
- * Adicione uma opção de configuração `clustermgtd` `ec2_instance_missing_max_count` para permitir uma quantidade e configurável de novas tentativas para eventuais instâncias de descrição do EC2 com instâncias executadas.

Alterações

- Atualize Slurm para 23.11.4 (de 23.02.7).
- Atualize o driver NVIDIA para a versão 535.154.05.
- Adicione suporte para Python 3.11, 3.12 na CLI do `pcluster` e `aws-parallelcluster-batch-cli`
- Crie interfaces de rede usando o índice da placa de rede da `NetworkCard`

`rdIndex` lista de Describel
nstances respostas do EC2,
em vez de repetir o alcance.
`MaximumNetworkCards`

- Falha na criação do cluster ao usar os tipos de instância P3, G3, P2 e G2 porque sua arquitetura de GPU não é compatível com os drivers Nvidia de código aberto (OpenRM) introduzidos como parte da versão 3.8.0.
- Atualize dependências de livros de receitas de terceiros: `nfs-5.1.2` (do `nfs-5.0.0`)
- Atualize o instalador EFA para `1.30.0`.
 - Driver EFA: `efa-2.6.0-1`
 - Configuração EFA: `efa-config-1.15-1`
 - Perfil da EFA: `efa-profile-1.6-1`
 - Libfabric-AWS: `libfabric-aws-1.19.0`
 - Núcleo RDMA: `rdma-core-46.0-1`
 - Abra o MPI: `openmpi40-aws-4.1.6-2` e `openmpi50-aws-5.0.0-11`

- Atualize o NICE DCV para a versão 2023.1-16388.
 - Servidor: 2023.1.16388-1
 - xdcv: 2023.1.565-1
 - gl: 2023.1.1047-1
 - web_viewer: 2023.1.16388-1

Correções de erros

- Corrija o problema que fazia com que o trabalho falhasse quando enviado como usuário do Active Directory a partir dos nós de login. O problema foi causado por uma configuração incompleta da integração com o Active Directory externo no nó principal.
- Refatore as políticas do IAM definidas no CloudFormation modelo parallelcluster-policies.yaml para evitar ParallelCluster falhas na implantação da API causadas por políticas que excedem os limites do IAM.
- Corrija o problema que fazia com que os nós de login falhassem na inicialização quando o nó principal demorava mais do que o

esperado para escrever as chaves.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster Lançada a versão 2024.02.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2024.02.0 da interface do usuário

8 de fevereiro de 2024

Alterações:

- Atualizado o ambiente de execução do Lambda para Python v3.9

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster Lançada a versão 2023.12.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2023.12.0 da interface do usuário.

21 de dezembro de 2023

Recursos:

- Foi adicionado suporte para implantação de PCUI com rede privada.
- Foi adicionada a possibilidade de aplicar opcionalmente um limite de permissões a cada função do IAM criada pelas infraestruturas PCUI e PCAPI
- Foi adicionada a possibilidade de aplicar opcionalmente um prefixo a cada função e política do IAM criada pela infraestrutura PCUI e PCAPI.
- Foi adicionado suporte para a ParallelCluster versão 3.8.0, sem paridade de recursos no assistente.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-uipacote](#) em GitHub.

[AWS ParallelCluster versão 3.8.0 lançada](#)

AWS ParallelCluster versão 3.8.0 lançada.

19 de dezembro de 2023

Melhorias:

- Adicione suporte para blocos de capacidade do EC2 para ML.
- Adicione suporte para Rocky Linux 8 conforme CustomAmi criado por meio do build-image processo. Nenhuma AMI pública oficial do AWS ParallelCluster Rocky8 Linux está disponível no momento.
- Adicione um Scheduling/ScalingStrategy parâmetro para controlar a estratégia de escalabilidade de cluster a ser usada ao iniciar instâncias do EC2 para os nós de computação do Slurm. Os valores possíveis são all-or-nothing greedy-all-or-nothing ,best-effort ,, com all-or-nothing sendo o padrão.
- Adicione um HeadNode/SharedStorageType parâmetro para usar o armazenamento EFS

em vez das exportações NFS do volume raiz do nó principal para recursos do sistema de arquivos compartilhado dentro do cluster: Intel ParallelCluster, Slurm e dados. /home Esse aprimoramento reduz a carga na rede do nó principal.

- Permita a montagem /home como um armazenamento compartilhado externo EFS ou FSx por meio da SharedStorage seção do arquivo de configuração.
- Adicione um novo parâmetro SlurmSettings/MungeKeySecretArn para permitir o uso de uma chave MUNGE externa definida pelo usuário do AWS Secrets Manager.
- Adicione um Monitoring/Alarms/Enabled parâmetro para ativar os Amazon CloudWatch Alarms para o cluster.
- Adicione alarmes do nó principal para monitorar as verificações de integridade do EC2, a utilização da CPU e o status geral do nó principal, e adicione-os ao

CloudWatch painel criado com o cluster.

- Adicione suporte para associações de repositório de dados ao usar `PERSISTENT_2` como um `FSx DeploymentType` for Lustre gerenciado.
- Adicione um `Scheduling/SlurmSettings/Database/DatabaseName` parâmetro para permitir que os usuários especifiquem um nome personalizado para o banco de dados no servidor de banco de dados a ser usado na contabilidade do Slurm.
- Crie `InstanceType` um parâmetro de configuração opcional ao configurar `CapacityReservationTarget/CapacityReservationId` no recurso computacional.
- Adicione a possibilidade de especificar um prefixo para as funções e políticas do IAM criadas pela AWS ParallelCluster API.
- Adicione a possibilidade de especificar um limite de permissões a ser aplicado às funções e políticas do

IAM criadas pela AWS
ParallelCluster API.

Alterações

- Atualize o Slurm para 23.02.7 (de 23.02.6).
- Atualize o driver NVIDIA para a versão 535.129.03.
- Atualize o CUDA Toolkit para a versão 12.2.2.
- Use drivers de GPU NVIDIA de código aberto (OpenRM) como módulo de kernel NVIDIA para Linux em vez de módulo de código fechado NVIDIA.
- Remova o suporte ao parâmetro de `all_or_nothing_batch` configuração no programa de retomada do Slurm, em favor da nova configuração do `Scheduling/ScalingStrategy` cluster.
- A convenção de nomenclatura de alarmes de cluster foi alterada para '[cluster-name] - [component-name] - [metric]'.
- Altere os tipos de volume padrão do EBS nas regiões do ADC de gp2 para gp3, tanto para os volumes raiz

quanto para os volumes adicionais.

- O limite de permissões opcionais para a AWS ParallelCluster API agora é aplicado a todas as funções do IAM criadas pela infraestrutura da API.
- Atualização do instalador EFA para 1.29.1.
- Driver EFA: efa-2.6.0-1
- Configuração EFA: efa-config-1.15-1
- Perfil da EFA: efa-profile-1.5-1
- Libfabric-AWS: libfabric-aws-1.19.0-1
- Núcleo RDMA: rdma-core-46.0-1
- Open MPI: openmpi40-aws-4.1.6-1
- Atualize o GDRCopy para a versão 2.4 em todos os sistemas operacionais compatíveis, exceto no Centos 7, onde a versão 2.3.1 é usada.
- Atualize aws-cfn-bootstrap para a versão 2.0-28.

- Adicione suporte para Python 3.10 em. `aws-parallelcluster-batch-cli`

Correções de erros

- Corrija a configuração de escalabilidade inconsistente após a reversão da atualização do cluster ao modificar a lista de tipos de instância declarados nos Recursos de computação.
- Corrija a geração de chaves SSH dos usuários ao alternar usuários sem privilégio de root em clusters integrados a um servidor LDAP externo por meio de arquivos de configuração de cluster.
- Corrija a desativação do modo de economia de energia do Slurm durante a configuração. `Scaledown IdleTime = -1`
- Corrija o caminho codificado para o diretório de instalação do Slurm no `update_slurm_database_password.sh` script do Slurm Accounting.

[AWS ParallelCluster versão
3.7.2 lançada](#)

AWS ParallelCluster versão
3.7.2 lançada.

25 de outubro de 2023

Alterações:

- Atualize o Slurm para 23.02.6.

[AWS ParallelCluster Lançada a versão 2023.10.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2023.10.0 da interface do usuário.

20 de outubro de 2023

Recursos:

- Foi adicionado suporte para ParallelCluster 3.7.2 com paridade de recursos no assistente limitada ao FSx File Cache e compatibilidade de agendamento baseada em memória com vários tipos de instância.

Correções de erros:

- Foi corrigido o problema que causava erros na interface do usuário quando a PCUI não tinha permissões para interagir com o Cost Explorer.

Melhorias

- Segurança aprimorada com a redução do TTL do token de acesso de 10 minutos para 5 minutos.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do

[aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster versão 3.7.1 lançada](#)

AWS ParallelCluster versão 3.7.1 lançada.

22 de setembro de 2023

Alterações:

- Atualize o Slurm para 23.02.5 (de 23.02.4).
 - Atualize o Pmix para 4.2.6 (de 3.2.3).
 - Atualize libjwt para 1.15.3 (de 1.12.0).
- Atualize o instalador do EFA para 1.26.1, corrigindo o problema de gravação de dados RDMA no P5.
 - Controlador da EFA: efa-2.5.0-1
 - Configuração EFA: efa-config-1.15-1
 - Perfil da EFA: efa-profile-1.5-1
 - Libfabric-AWS: libfabric-aws-1.18.2-1
 - Núcleo EDMA: rdma-core-46.0-1
 - Abra o MPI: openmpi40-aws-4.1.5-4

[AWS ParallelCluster versão 3.7.0 lançada](#)

AWS ParallelCluster versão 3.7.0 lançada.

30 de agosto de 2023

Melhorias:

- Support a configuração de prioridades de nós estáticos e dinâmicos em recursos computacionais usando um arquivo YAML AWS ParallelCluster de configuração.
- Adicionado suporte para Ubuntu 22. As chaves RSA não são suportadas por padrão.
- Adicionada a configuração de fila `JobExclusiveAllocation` para alocar nós em uma partição exclusivamente para um único trabalho a qualquer momento.
- Permita substituir o `aws-parallelcluster-node` pacote no momento da criação e atualização do cluster. Para o nó principal, isso se aplica à atualização do cluster. Útil somente para fins de desenvolvimento.
- Evita iniciar o servidor NFS nos nós de computação.

- Adicionado suporte para nós de login.
- Permite o agendamento baseado em memória quando vários tipos de instância são especificados para um recurso de computação do Slurm.
- Adicionado suporte para montar o Amazon File Cache existente como armazenamento compartilhado.

Alterações:

- Atribuído aos nós dinâmicos do Slurm uma prioridade (peso) de 1000 por padrão. Ao fazer isso, o Slurm pode priorizar nós estáticos ociosos ao invés de nós dinâmicos ociosos.
- Faça com que os `aws-parallelcluster-node daemons` lidem apenas com partições AWS ParallelCluster gerenciadas Slurm.
- Aumento do intervalo de sondagem de vigilância (`watchdog`) `EFS-utils` para 10 segundos. Essa alteração se aplica quando `EncryptionInTransit` está definida como `true`,

que é a única condição que faz com que o watchdog seja executado.

- Atualização do instalador do EFA para 1.25.1.
 - Driver EFA: efa-2.5.0-1 (era efa-2.1.1g)
 - Efa-config: efa-config-1.15-1 (era efa-config-1.13-1)
 - Perfil EFA: efa-profile-1.5-1 (sem alteração)
 - Libfabric-aws: libfabric-aws-1.18.1-0 (de libfabric-aws-1.17.1-1)
 - Núcleo RDMA: rdma-core-46.0-1 (era rdma-core-43.0-1)
 - Open MPI: openmpi40-aws-4.1.5-4 (era openmpi40-aws-4.1.5-1)
- Atualização do Slurm para a versão 23.02.4.
- Altere o valor padrão de Imds/ ImdsSupport de v1.0 para v2.0.
- Descontinuação do Ubuntu 18.
- Atualização do tamanho padrão do volume raiz para

40 GB para considerar os limites do Centos 7.

- Restrita a permissão no arquivo `/tmp/wait_condition_handle.txt` dentro do nó principal para que somente o root possa lê-lo.
- Criado um arquivo JSON de mapeamento `partition-nodelist` do Slurm para ser usado pelos daemons do pacote de nós para reconhecer partições e listas de nós do Slurm gerenciadas por PC.
- Atualização do driver NVIDIA para a versão 535.54.03.
- Atualização da biblioteca CUDA para a versão 12.2.0.
- Atualize o NVIDIA Fabric Manager para `nvidia-fabricmanager-535`.
- Atualização do ARM PL para a versão 23.04.1 somente para Ubuntu 22.04.
- Atualização do NICE DCV para a versão 2023.0-15487 .
 - Servidor: 2023.0.15487-1
 - xdcv: 2023.0.551-1
 - gl: 2023.0.1039-1

- `web_viewer: 2023.0.15`
`487-1`

Correções de erros:

- Adicionada validação ao valor `Scaledown IdleTime` para evitar definir um valor menor que `-1`.
- Corrigida a falha na criação do cluster com a AMI do Ubuntu Deep Learning em instâncias de GPU com DCV ativado.
- Corrija o problema que fazia com que políticas pendentes do IAM fossem criadas ao criar um provedor de recursos ParallelCluster CloudFormation personalizado com `CustomLambdaRole`
- Corrigido um problema que estava causando o desalinhamento do nome DNS dos nós de computação em instâncias com várias interfaces de rede, ao usar `SlurmSettings/Dns/UseEc2Hostnames equals to True`

Para obter detalhes sobre as alterações, consulte os

CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[Versão somente para documentação](#)

AWS ParallelCluster guia do usuário específico da versão 3 publicado.

17 de julho de 2023

Versão somente para documentação:

- AWS ParallelCluster a versão 3 tem seu próprio guia de usuário separado.

[AWS ParallelCluster versão
3.6.1 lançada](#)

AWS ParallelCluster versão
3.6.1 lançada.

5 de julho de 2023

Alterações:

- Evita a duplicação de nós vistos pelo `clusterimgtd` se os nós de computação forem adicionados a várias partições Slurm.

Correções de erros:

- Remove a codificação rígida do nome do dispositivo do volume raiz (`/dev/sda1` e `/dev/xvda`) e recupera-as das AMIs usadas durante `create-cluster`.
- Corrija a falha na criação do cluster ao usar o recurso CloudFormation personalizado com `ElasticIp` definido como `True`.
- Corrija falhas na criação e atualização do cluster ao usar um recurso AWS CloudFormation personalizado com grandes arquivos de configuração.
- Corrige um problema que impedia a desativação da proteção `pttrace` no Ubuntu e que não permitia o Cross

Memory Attach (CMA) no libfabric.

- Corrige a lógica rápida de failover de capacidade e insuficiente ao usar vários tipos de instância e nenhuma instância ser retornada.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster Lançada a versão 2023.06.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2023.06.0 da interface do usuário.

7 de junho de 2023

Alterações:

- Atualizou a versão padrão AWS ParallelCluster da API para 3.6.0.

Correções de erros:

- Implantação interrompida corrigida na região AWS GovCloud (Oeste dos EUA).
- O painel dividido agora carrega corretamente os detalhes do cluster após o início da criação.

Observações:

- O recurso de monitoramento de custos não está disponível em AWS GovCloud (US) Regions.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster versão 3.6.0 lançada](#)

AWS ParallelCluster versão 3.6.0 lançada.

22 de maio de 2023

Documentação:

- Adicionada documentação para o [API da biblioteca Python do AWS ParallelCluster](#).

Melhorias:

- Adição de suporte para RHEL8.
- Adicione um [recurso AWS CloudFormation personalizado](#) para criar e gerenciar clusters com CloudFormation.
- Adicione suporte para [personalizar a Slurm configuração do cluster no arquivo](#) YAML AWS ParallelCluster de configuração.
- Compilação Slurm com suporte para LUA.
- Aumenta o limite do número máximo de filas por cluster de 10 para 50. Cada fila pode ter até 50 recursos de computação. Cada cluster pode ter até 50 recursos de computação.

- Adiciona suporte para especificar uma sequência de vários [scripts de ação personalizados](#) para um evento configurado em parâmetros `OnNodeStart`, `OnNodeConfigured`, e `OnNodeUpdated`.
- Adiciona uma nova seção de configuração `HealthChecks / Gpu`, para aplicar verificações de integridade da GPU em um nó de computação antes da execução de um trabalho.
- Adiciona suporte para `Tags` em `SlurmQueues` e na configuração `SlurmQueues / ComputeResources`.
- Adiciona suporte para [DetailedMonitoring](#) na configuração `Monitoring`.
- Adicione `mem_used_percent` `disk_used_percent` métricas para a memória do nó principal e o rastreamento da utilização do disco do volume raiz no AWS ParallelCluster [CloudWatch painel](#) e configure alarmes para monitorar essas métricas.

- Adiciona suporte à [rotação de logs](#) para registros gerenciados do AWS ParallelCluster .
- [Acompanhe os erros comuns do nó de computação e o maior tempo de inatividade do nó dinâmico no painel. CloudWatch](#)
- Garante que o DCV Authenticator Server use pelo menos o protocolo TLS-1.2 ao criar o soquete SSL.
- Instalação do pacote [NVIDIA Data Center GPU Manager \(DCGM\)](#) em todos os sistemas operacionais compatíveis, exceto aarch64 centos7 e alinux2.
- Carrega o módulo do kernel [nvidia-vm](#) por padrão para fornecer a funcionalidade de Memória Virtual Unificada (UVM) ao driver CUDA.
- Instala o [NVIDIA Persistence Daemon](#) como um serviço do sistema.

Alterações:

- Atualização do Slurm para a versão 23.02.2 (era versão 22.05.8).
- Atualização do munge para a versão 0.5.15 (era versão 0.5.14).
- Define o `TreeWidth` do Slurm para 30.
- Define as configurações `prolog` e `epilog` do Slurm para o diretório de destino `/opt/slurm/etc/scripts/prolog.d/` e `/opt/slurm/etc/scripts/epilog.d/`, respectivamente.
- Define `BatchStartTimeout` do Slurm para um máximo de 3 minutos para executar scripts `Prolog` durante o registro do nó de computação.
- Aumente o padrão `RetentionInDays` dos `CloudWatch` registros de 14 para 180 dias.
- Atualização do instalador do EFA para 1.22.1.
 - `Dkms`: 2.8.3-2
 - `Driver EFA`: `efa-2.1.1g` (sem alteração)
 - `Efa-config`: `efa-config-1.13-1` (sem alteração)

- Perfil EFA: efa-profile-1.5-1 (sem alteração)
- Libfabric-aws: libfabric-aws-1.17.1-1 (era libfabric-aws-1.17.0-1)
- Núcleo RDMA: rdma-core-43.0-1 (sem alteração)
- Open MPI: openmpi40-aws-4.1.5-1 (sem alteração)
- Atualização da versão do cliente Lustre para a 2.12 no Amazon Linux 2. O cliente Lustre 2.12 foi instalado no Ubuntu 20.04, 18.04 e CentOS >= 7.7.
- Atualização da versão do cliente Lustre para a 2.10.8 no CentOS 7.6.
- Atualização do driver NVIDIA para a versão 470.182.03 (era versão 470.141.03).
- Atualização do NVIDIA Fabric Manager para a versão 470.182.03 (era versão 470.141.03).
- Atualização do NVIDIA CUDA Toolkit para a versão 11.8.0 (era versão 11.7.1).

- Atualização da amostra NVIDIA CUDA para a versão 11.8.0.
- Biblioteca Intel MPI atualizada para a Versão 2021 atualização 9 (era a versão 2021 atualização 6). Para obter mais informações, consulte [Intel® MPI Library 2021 Update 9](#).
- Atualização do NICE DCV para a versão 2023.0-15022 (era a versão 2022.2-14521).
 - Servidor: 2023.0.15022-1 (era versão 2022.2-14521-1).
 - xdcv: 2023.0.547-1 (era versão 2022.2.519-1).
 - gl: 2023.0.1027-1 (era versão 2022.2.1012-1).
 - web_viewer: 2023.0.15022-1 (era versão 2022.2.14521-1).
- Atualização do aws-cfn-bootstrap para a versão 2.0-24.
- Atualize a imagem usada pelo CodeBuild ambiente ao criar imagens de contêiner para AWS Batch clusters:

- `aws/codebuild/amazonlinux2-x86_64-standard:4.0` (de `aws/codebuild/amazonlinux2-x86_64-standard:3.0`).
- `aws/codebuild/amazonlinux2-aarch64-standard:2.0` (era `aws/codebuild/amazonlinux2-aarch64-standard:1.0`).

Correções de erros:

- Corrige os validadores de grupos de segurança de rede Amazon EFS e Amazon FSx para evitar relatórios de erros falsos.
- Corrige a falta de tags em recursos criados pelo Image Builder durante a operação `build-image` .
- Corrige a política de atualização `MaxCount` para sempre realizar comparações numéricas na propriedade de `MaxCount`.
- Corrige o alinhamento de IP em instâncias de nós de computação com várias placas de rede.
- Corrige a substituição de `StoragePass` no

`slurm_parallelcluster_slurmdbd.conf`
quando uma atualização de parâmetros de fila é executada e as configurações contábeis do Slurm não são atualizadas.

- Corrige o problema que faz com que grupos de segurança pendentes sejam criados ao criar um cluster com um sistema de arquivos EFS existente.
- Corrige o problema que faz com que o daemon `cfn-hup` falhe ao ser reiniciado.
- Considera nós dinâmicos com sinalizador `INVALID_REGION` como falhas de bootstrap para o modo protegido Slurm. Os nós estáticos com falha no registro Slurm já são tratados como falhas de bootstrap após o `node_replacement_timeout`.

Para obter detalhes sobre as alterações, consulte os `CHANGELOG` arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster Lançada a versão 2023.05.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2023.05.0 da interface do usuário.

16 de maio de 2023

Melhorias:

- A partir da AWS ParallelCluster versão 3.6.0, adicione suporte para o RHEL 8.
- Adiciona o monitoramento de custos do cluster.
- A partir da AWS ParallelCluster versão 3.6.0, aumente as cotas de recursos de fila e computação.

Alterações:

- Melhora a interface de usuário do assistente de criação de clusters.
- Aumentou a velocidade da implantação da AWS ParallelCluster interface do usuário.
- Melhora a interface para adicionar um novo usuário.
- Por padrão, as filas estão na sub-rede do nó principal.

Correções de erros:

- Muda para a região correta após a conclusão da criação do cluster.
- Corrige a exibição do indicador de carregamento no recurso "Editar cluster".
- Corrija a criação do cluster quando a SnapshotId propriedade do EBS for removida.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster Lançada a versão 2023.04.0 da interface do usuário](#)

AWS ParallelCluster Lançada a versão 2023.04.0 da interface do usuário.

17 de abril de 2023

Melhorias:

- Design reformulado do assistente de criação de clusters.
- Design reformulado da página de logs do cluster.
- Adiciona uma configuração de nome personalizada para armazenamento compartilhado.
- Adiciona várias opções de armazenamento ao adicionar armazenamento a um cluster.
- Adiciona suporte `DeletionPolicy` para Amazon EFS e FSx para Lustre.
- Adiciona uma opção `ImdsSupport` na configuração do cluster.
- Adiciona suporte aos tipos de instância C7.
- Tutorial [Revertendo para uma versão anterior do documento do AWS Systems Manager](#) adicionado.

Alterações:

- Configuração de cluster YAML de até 1 MB de tamanho.
- O usuário não é desconectado devido a uma autorização com credenciais temporárias do Boto3 IAM.
- Opções de threads múltiplas desativadas quando uma instância de HPC é selecionada.
- Removida a desativação da reversão na página de criação do cluster.
- O usuário é impedido de usar a AWS ParallelCluster interface do usuário até que as informações necessárias sejam fornecidas.
- Até 10 filas podem ser adicionadas.
- O documento `SSM-SessionManagerRunShell` não é sobrescrito durante a instalação da interface do usuário do AWS ParallelCluster .

Correções de erros:

- Corrige o link de redefinição de senha quebrado.

- Corrige quebra de delete stack causada pelo EcrPrivateRepository não estar vazio
- Problema de inicialização corrigido da caixa de seleção Gerar chaves SSH na seção Propriedades de gerenciamento de vários usuários.
- Correção da pane causada por um trabalho com propriedades indefinidas.
- Correção de configurações do SCRATCH FSx.
- Correção do botão de Iniciar e Parar instâncias, ainda ativado após ser clicado uma vez.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster-ui](#) pacote em GitHub.

[AWS ParallelCluster versão 3.5.1 lançada](#)

AWS ParallelCluster versão 3.5.1 lançada.

29 de março de 2023

Melhorias:

- Adiciona um [executável autônomo do instalador da CLI pcluster](#).

Alterações:

- Atualização do instalador EFA para 1.22.0.
 - Driver EFA: efa-2.1.1g (era efa-2.1.1-1)
 - Efa-config: efa-config-1.13-1 (era efa-config-1.12-1)
 - Perfil EFA: efa-profile-1.5-1 (sem alteração)
 - Libfabric-aws: libfabric-aws-1.17.0-1 (era libfabric-aws-1.16.1amzn3.0-1)
 - Núcleo RDMA: rdma-core-43.0-1 (sem alteração)
 - Open MPI: openmpi40-aws-4.1.5-1 (era openmpi40-aws-4.1.4-3)

Atualização do NICE DCV
para a versão 2022.2-14
521 .

- Servidor: 2022.2.14
521-1
- xdcv: 2022.2.519-1
- gl: 2022.2.1012-1
- web_viewer: 2022.2.14
521-1

Correções de erros:

- Corrige possíveis falhas de inicialização de nós causadas pela correspondência de padrões entre `MountDir` e `/etc/exports` ao remover volumes compartilhados do Amazon EBS como parte de uma atualização de cluster.
- Correção para evitar o truncamento do arquivo de log `compute_console_output` em cada iteração `clustermgtd` .

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluste](#)

[r-cookbookaws-parallelcluster-
node](#) GitHub

[AWS ParallelCluster versão 3.5.0 lançada](#)

AWS ParallelCluster versão 3.5.0 lançada.

20 de fevereiro de 2023

Melhorias:

- Acesso e gerenciamento de clusters com a [interface do usuário do AWS ParallelCluster](#).
- Adicione AWS ParallelCluster políticas versionadas em um CloudFormation modelo que você pode referenciar em suas cargas de trabalho.
- Adicione uma biblioteca AWS ParallelCluster Python que você possa usar com seu próprio código.
- Adicione o registro da saída do console do nó de computação à Amazon CloudWatch em caso de falha no bootstrap do nó de computação.
- Adiciona o campo de falhas contendo o código da falha e o motivo da saída do `describe-cluster` quando a criação do cluster falhar.
- Adiciona validadores para evitar a injeção maliciosa de

strings ao chamar o módulo de subprocesso.

- Falha na criação do cluster se o status do cluster mudar para PROTECTED durante o provisionamento de nós estáticos.

Alterações:

- Atualização do Slurm versão 22.05.8 (era versão 22.05.7).
- Atualização do instalador EFA para 1.21.0.
 - Driver EFA: efa-2.1.1-1 (era efa-2.1)
 - Efa-config: efa-config-1.12-1 (era efa-config-1.11-1)
 - Perfil EFA: efa-profile-1.5-1 (sem alteração)
- Libfabric-aws: libfabric-aws-1.16.1amzn3.0-1 (de libfabric-aws-1.16.1)
- Núcleo RDMA: rdma-core-43.0-1 (era rdma-core-43.0-2)
- Open MPI: openmpi40-aws-4.1.4-3 (sem alteração)

- Torna os logs do controlador Slurm mais detalhados e ativa o registro adicional para o plug-in de economia de energia do Slurm.

Correções de erros:

- Corrige a criação do banco de dados do cluster verificando se o nome do cluster não passa de 40 caracteres quando a contabilidade do Slurm está ativada.
- Corrige um problema em `clustermgtd` que fazia com que os nós de computação, reinicializados por meio de Slurm, fossem substituídos se as verificações de status da instância do EC2 falhassem.
- Corrige um problema que impedia o lançamento de nós de computação, com reservas de capacidade compartilhadas por outras contas, devido a uma política de IAM incorreta no nó principal.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), [aws-parallelcluster-cookbook](#), [aws-parallelcluster-cookbook](#)

[lelcluster-node](#) e os pacotes em. [aws-parallelcluster-ui](#) GitHub

[AWS ParallelCluster versão 3.4.1 lançada](#)

AWS ParallelCluster versão 3.4.1 lançada.

13 de janeiro de 2023

Correções de erros:

- Corrige um problema do programador do Slurm que poderia causar a aplicação incorreta de atualizações em seu registro interno de nós de computação. Como resultado desse problema, as instâncias do EC2 podem ficar indisponíveis ou serem apoiadas por um tipo de instância incorreto.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbook](#) [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão 3.4.0 lançada](#)

AWS ParallelCluster versão 3.4.0 lançada.

22 de dezembro de 2022

Melhorias:

- Adiciona suporte para lançar nós em várias zonas de disponibilidade para aumentar a disponibilidade da capacidade.
- Adiciona suporte para especificar várias sub-redes para cada fila para aumentar a disponibilidade da capacidade.
- Adiciona um novo parâmetro de configuração em [Iam](#) / [ResourcePrefix](#) para especificar um prefixo para o caminho e o nome dos recursos do IAM criados pelo AWS ParallelCluster.
- Adicione a nova seção de configuração [DeploymentSettings](#) / [LambdaFunctionsVpcConfig](#) para especificar a configuração Vpc usada pelas funções AWS ParallelCluster Lambda.
- Adiciona a capacidade de especificar um script personalizado para ser

executado no nó principal durante uma atualização do cluster. O script pode ser especificado com [HeadNode / CustomActions / OnNodeUpdated](#) ao usar Slurm como programador.

Alterações:

- Remove a criação de destinos de montagem do Amazon EFS para sistemas de arquivos existentes.
- Monta sistemas de arquivos EFS usando o `amazon-efs-utils`. Os sistemas de arquivos EFS podem ser montados usando criptografia em trânsito e um usuário autorizado pelo IAM.
- Instala o `stunnel` 5.67 no CentOS7 e no Ubuntu para oferecer suporte à criptografia EFS em trânsito.
- Atualização do instalador EFA para 1.20.0 (era 1.18.0).
 - Driver EFA: `efa-2.1` (era `efa-1.16.0-1`)
 - Efa-config: `efa-config-1.11-1` (sem alteração)

- Perfil EFA: efa-profile-1.5-1 (sem alteração)
- Libfabric-aws: libfabric-aws-1.16.1 (era libfabric-aws-1.16.0~amzn4.0-1)
- Núcleo RDMA: rdma-core-43.0-2 era (rdma-core-41.0-2)
- Open MPI: openmpi40-aws-4.1.4-3 era (openmpi40-aws-4.1.4-2)
- Atualiza o Slurm para a versão 22.05.7 (era 22.05.5).
- Atualiza o Python para 3.9.16 e 3.7.16 (era 3.9.15 e 3.7.13).
- Com o Slurm 22.05.7, os nós dinâmicos em status IDLE+CLOUD+COMPLETING+POWER_DOWN+NOT_RESPONDING deixam de ser considerados não íntegros.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluste](#)

[r-cookbookaws-parallelcluster-
node](#) GitHub

[AWS ParallelCluster versão
3.3.1 lançada](#)

AWS ParallelCluster versão
3.3.1 lançada.

2 de dezembro de 2022

Alterações:

- As AMIs oficiais AWS ParallelCluster do produto agora estão disponíveis após a suspensão do uso do Amazon EC2 em dois anos.
- Aumente o tamanho da memória da AWS ParallelCluster API Lambda para 2048 para reduzir as penalidades de inicialização a frio e evitar tempos limite.

Correções de erros:

- Evita a substituição de sistemas de arquivos FSx para Lustre gerenciados e perda de dados em atualizações de cluster que incluam alterações no ID de sub-rede da frota de computação.
- [SharedStorage](#) DeletionPolicy aplica-se às ações de atualização do cluster.

Para obter detalhes sobre as mudanças, consulte o CHANGELOG arquivo do pacote [aws-parallelcluster](#) em GitHub

[AWS ParallelCluster somente documentação \(nota hpc6id\)](#)

AWS ParallelCluster atualização somente de documentação

2 de dezembro de 2022

- AWS ParallelCluster não suporta o tipo de instância hpc6id para a [HeadNode](#) configuração / [InstanceType](#)

[AWS ParallelCluster versão
3.1.5 lançada](#)

AWS ParallelCluster versão
3.1.5 lançada.

16 de novembro de 2022

Melhorias:

- Corrige o problema do Slurm que impede o encerramento de nós ociosos.
- Atualização do instalador EFA para 1.18.0
 - Driver EFA: efa-1.16.0-1
 - Efa-config: efa-config-1.11-1 (era efa-config-1.9-1)
 - Perfil EFA: efa-profile-1.5-1 (sem alteração)
 - Libfabric-aws: libfabric-aws-1.16.0~amzn4.0-1 (era libfabric-1.13.2)
 - Núcleo RDMA: rdma-core-41.0-2 (era rdma-core-37.0)
 - Open MPI: openmpi40-aws-4.1.4-2 (era openmpi40-aws-4.1.1-2)

Alterações:

- Adicione `lambda:ListTags` e `lambda:UntagResource` ao `ParallelClusterUserRole` usado pela pilha de AWS ParallelCluster APIs para uma atualização do cluster.
- Biblioteca Intel MPI atualizada para a versão 2021 atualização 6 (atualizada a partir da versão 2021 atualização 4). Para obter mais informações, consulte [Intel® MPI Library 2021 Update 6](#).
- Atualização do driver NVIDIA para a versão 470.141.03 (era 470.103.01).
- Atualização do NVIDIA Fabric Manager para a versão 470.141.03 (era 470.103.01).

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão 3.3.0 lançada](#)

AWS ParallelCluster versão 3.3.0 lançada.

2 de novembro de 2022

Melhorias:

- Adiciona suporte à configuração de alocação de várias instâncias para um recurso de computação ao usar Slurm como programador. Para ter mais informações, consulte [Alocação a vários tipos de instância com o Slurm](#).
- Adiciona suporte para adição e remoção do [SharedStorage](#) com uma atualização de cluster, usando uma configuração atualizada. Para ter mais informações, consulte [Armazenamento compartilhado](#).
- Adiciona um novo parâmetro de configuração DeletionPolicy para configurações de armazenamento compartilhado [Efs](#) e [FsxLustre](#) para oferecer suporte à retenção de armazenamento.
- Adiciona suporte para contabilidade Slurm com o novo parâmetro de

configuração [Scheduling / SlurmSettings / Database](#). Para ter mais informações, consulte [Slurmcontabilidade com AWS ParallelCluster](#).

- Adiciona suporte para reservas de capacidade sob demanda e grupos de recursos de reserva de capacidade. Para ter mais informações, consulte [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#).
- Adição de um novo parâmetro de configuração para especificar a versão do IMDS a ser suportada em um cluster ou infraestrutura de imagem de compilação nas configurações de cluster, [Imds / ImdsSupport](#), e compilação, [Imds / ImdsSupport](#).
- Adiciona suporte para [Networking / PlacementGroup](#) na seção [SlurmQueues / ComputeResources](#).
- Adiciona suporte para instâncias com várias interfaces de rede limitadas a apenas uma ENI por dispositivo.

- Melhora a validação da rede para sistemas de arquivos externos do Amazon EFS verificando o bloco CIDR no grupo de segurança anexado.
- Adiciona um validador para verificar se os tipos de instância configurados são compatíveis com grupos de posicionamento.
- Configura os threads NFS como min (256, max (8, num_cores * 4)) para garantir melhor estabilidade e desempenho.
- Move a instalação do NFS no momento da compilação para reduzir o tempo de configuração.
- Ative a criptografia do lado do servidor para o tópico do EcrImageBuilder SNS criado durante a implantação da AWS ParallelCluster API e usado para notificar eventos de criação de imagens do docker.

Alterações:

- Altera o comportamento de [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#) / Enabled. Agora, ele cria

um grupo de posicionamento gerenciado exclusivo para cada recurso de computação, em vez de um único grupo de posicionamento gerenciado para todos os recursos de computação.

- Adiciona suporte para [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#) / [Name](#) como o método de nomenclatura preferido.
- Move as tags do nó principal do Launch Template para a definição da instância para evitar a substituição do nó principal nas atualizações das tags.
- Desativa o multithreading por meio de um script executado por `cloud-init` e não por meio do `CpuOptions` definido no modelo de inicialização.
- Atualiza o Python para a versão 3.9 e o NodeJS para a versão 16 na infraestrutura da API, no contêiner do Docker da API e nos recursos do Lambda do cluster.
- Remove o suporte para o Python 3.6 no `aws-parallel`

lelcluster-batch-cl

- Atualiza o Slurm para a versão 22.05.5 (era 21.08.8-2).
- Atualização do driver NVIDIA para a versão 470.141.03 (era 470.129.06).
- Atualização do NVIDIA Fabric Manager para a versão 470.141.03 (era 470.129.06).
- Atualização do NVIDIA CUDA Toolkit para a versão 11.7.1 (from 11.4.4).
- Atualize o Python usado em AWS ParallelCluster virtualenvs de para. 3.7.13 3.9.15
- Atualização do instalador EFA para a versão 1.18.0.
 - Driver EFA: efa-1.16.0-1 (sem alteração)
 - Efa-config: efa-config-1.11-1 (from efa-config-1.10-1)
 - Perfil EFA: efa-profile-1.5-1 (sem alteração)
 - Libfabric-aws: libfabric-aws-1.16.0~amzn4.0-1 (era

- `libfabric-aws-1.16.0~amzn2.0-1`)
- Núcleo RDMA: `rdma-core-41.0-2` (era `rdma-core-37.0`)
- Open MPI: `openmpi40-aws-4.1.4-2` (era `openmpi40-aws-4.1.1-2`)
- Atualização do NICE DCV para a versão `2022.1-13300` (era `2022.0-12760`).
- Ativa a supressão do `SingleSubnetValidator` para Queues.
- Não substitui os nós do DRAIN quando os nós estiverem no estado `COMPLETING` em que o `Epilog` ainda pode estar em execução.

Correções de erros:

- Corrige a validação do parâmetro de filtros no `AWS ParallelCluster ListClusterLogStreams` comando para falhar quando filtros incorretos são passados.
- Corrija a validação do parâmetro [SharedStorage](#) /[EfsSettings](#) para

falhar na validação quando `FileSystemId` for especificado junto com outros [EfsSettings](#) parâmetros [SharedStorage](#) //.

Anteriormente, `FileSystemId` não estava incluído.

- Corrige a atualização do cluster ao alterar a ordem de [SharedStorage](#) junto com outras alterações na configuração.
- Correção `UpdateParallelClusterLambdaRole` na AWS ParallelCluster API para a qual fazer upload de registros `CloudWatch`.
- Corrige que o Cinc não usa o pacote de certificados CA local ao instalar pacotes antes que qualquer livro de receitas seja executado.
- Corrige um travamento na atualização do Ubuntu com `pcluster build-image` quando `Build:UpdateOsPackages:Enabled:true` está definido.
- Corrige a análise da configuração do cluster YAML falhando nas chaves duplicadas.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbook](#)[aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster somente documentação, referência de API adicionada.](#)

AWS ParallelCluster atualização somente de documentação

27 de outubro de 2022

- A versão 3 [Referência da API do AWS ParallelCluster](#) foi adicionada à documentação.

[AWS ParallelCluster versão
3.2.1 lançada](#)

AWS ParallelCluster versão
3.2.1 lançada.

3 de outubro de 2022

Melhorias:

- Melhora a lógica para associar as tabelas de roteamento do host às diferentes placas de rede para oferecer melhor suporte às instâncias do EC2 com várias NICs.

Alterações:

- Atualização do driver NVIDIA para a versão 470.141.03.
- Atualização do NVIDIA Fabric Manager para a versão 470.141.03.
- Desativa as tarefas de trabalho man-db e mlocate do cron, que podem ter um impacto negativo no desempenho do nó.
- Atualiza a Biblioteca Intel MPI para 2021.6.0.602.
- Atualiza o Python da versão 3.7.10 para a 3.7.13 em resposta a esse risco de segurança.

Correções de erros:

- Evita falhas em `DescribeCluster` quando a configuração do cluster não estiver disponível.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão 3.2.0 lançada](#)

AWS ParallelCluster versão 3.2.0 lançada.

27 de julho de 2022

Melhorias:

- Adiciona suporte para [agendamento baseado em memória](#) no Slurm.
 - Configura a memória real dos nós de computação na configuração de cluster do Slurm.
 - Adiciona o novo parâmetro de configuração [Scheduling](#) / [SlurmSettings](#) / [EnableMemoryBasedScheduling](#) para habilitar o agendamento baseado em memória no Slurm.
 - Adiciona o novo parâmetro de configuração [Scheduling](#) / [SlurmQueues](#) / [ComputeResources](#) / [SchedulableMemory](#) para substituir o valor padrão da memória vista pelo programador nos nós de computação.
- Melhora a flexibilidade nas atualizações de configuração do cluster para evitar

a parada e o início do cluster inteiro sempre que possível. Adiciona o novo parâmetro de configuração [Scheduling / SlurmSettings / QueueUpdateStrategy](#) para definir a estratégia preferida a ser usada quando os nós de computação precisarem de uma atualização e substituição da configuração.

- Melhora o mecanismo de failover dos recursos de computação disponíveis ao resolver problemas de capacidade insuficiente com instâncias do EC2. [Desativa os nós de computação por um período de tempo configurável](#) quando a inicialização de um nó falhar devido à capacidade insuficiente.
- Adiciona suporte para montar sistemas de arquivo [FSx for ONTAP](#) e [FSx for OpenZFS](#) existentes.
- Adiciona suporte para montar várias instâncias dos sistemas de arquivos [Amazon Elastic File Systems](#), [FSx para Lustre](#), [FSx para ONTAP](#) e [FSx para Open ZFS](#) existentes.

- Adiciona suporte para o [tipo de implantação Persistente_2 do FSx para Lustre](#) ao criar um novo sistema de arquivos.
- Solicita que o usuário habilite o EFA para os tipos de instância compatíveis ao usar o assistente do [pcluster configure](#).
- Adiciona suporte para reinicializar nós de computação usando o Slurm.
- Melhora o tratamento dos estados de energia do Slurm para também considerar o desligamento manual dos nós.
- Instala o NVIDIA GDRCopy 2.3 nas AMIs do produto para permitir a cópia da memória da GPU de baixa latência.

Alterações:

- Atualização do instalador EFA para a versão 1.17.2
 - EFA driver: efa-1.16.0-1
 - EFA configuration: efa-config-1.10-1
 - Perfil EFA: efa-profile-1.5-1

- Libfabric: libfabric-aws-1.16.0~amzn2.0-1
- RDMA core: rdma-core-41.0-2
- Open MPI: openmpi40-aws-4.1.4-2
- Atualização do NICE DCV para a versão 2022.0-12760.
- Atualização do driver NVIDIA para a versão 470.129.06.
- Atualização do NVIDIA Fabric Manager para a versão 470.129.06.
- Altera os tipos de volume padrão do EBS de gp2 para gp3 nos volumes raiz e adicionais.
- Alterações nos sistemas de arquivos FSx for Lustre criados por: AWS ParallelCluster
 - Altera o tipo de implantação padrão para Scratch_2 .
 - Altere a versão do servidor Lustre para 2.12.
- Não exige que [Placement Group](#) / [Enabled](#) seja definido como true ao

passar um Placement Group / Id existente.

- Não permite definir PlacementGroup / Id quando Placement Group / Enabled está explicitamente definido como false.
- Adiciona uma tag parallelcluster:cluster-name a todos os recursos criados pelo AWS ParallelCluster.
- Adicione lambda:ListTags e use lambda:UpdateTagResource pela ParallelClusterUserRoleRole pilha de AWS ParallelCluster APIs para atualização do cluster.
- Restringe o acesso IPv6 para IMDS somente aos usuários administradores raiz e do cluster, quando o parâmetro de configuração HeadNode / Imds / Secured estiver habilitado.
- Com uma AMI personalizada, use o tamanho do volume raiz da AMI em vez do ParallelCluster padrão de 35 GiB. O valor pode ser alterado no arquivo de configuração do cluster.

- Desabilitação automática da frota de computação quando o parâmetro de configuração `Scheduling / SlurmQueues / ComputeResources / SpotPrice` é inferior ao preço mínimo de atendimento de solicitação spot.
- Mostra valores `requested_value` e `current_value` no conjunto de alterações ao adicionar ou remover uma seção durante uma atualização.
- Desativa o serviço `aws-ubuntu-eni-helper`, disponível nas AMIs do Deep Learning, para evitar conflitos com `configure_nw_interface.sh` ao configurar instâncias com várias placas de rede.
- Remove o suporte para o Python 3.6.
- Define a MTU como 9001 para todas as interfaces de rede ao configurar instâncias com várias placas de rede.
- Remove o ponto final ao configurar o FQDN do nó de computação.
- Gerencia nós estáticos em `POWERING_DOWN`.

- Não substitui o nó dinâmico em `POWER_DOWN` , pois os trabalhos ainda podem estar em execução.
- Reinicia os daemons `clustermgtd` e `slurmctld` no momento da atualização do cluster somente quando os parâmetros `Scheduling` forem atualizados na configuração do cluster.
- Atualização dos arquivos de manutenção `slurmctld` e `slurmd` do `systemd`
- Restringe o acesso IPv6 para IMDS somente aos usuários administradores raiz e do cluster, quando o parâmetro de configuração `HeadNode / Imds / Secured` estiver habilitado.
- Define a configuração do Slurm `AuthInfo=cred_expire=70` para reduzir o tempo que os trabalhos necessários devem esperar antes de serem reiniciados quando os nós não estiverem disponíveis.
- Atualização de dependências de manuais de instruções (cookbook) de terceiros:
 - apt-7.4.2 (era apt-7.4.0)

- line-4.5.2 (era line-4.0.1)
- openssh-2.10.3 (era openssh-2.9.1)
- pyenv-3.5.1 (era pyenv-3.4.2)
- selinux-6.0.4 (era selinux-3.1.1)
- yum-7.4.0 (era yum-6.1.1)
- yum-epel-4.5.0 (era yum-epel-4.1.2)

Correções de erros:

- Corrija o comportamento padrão para ignorar as etapas de AWS ParallelCluster validação e teste ao criar uma AMI personalizada.
- Corrige o vazamento do identificador do arquivo em `computemgtd`.
- Corrija a condição de corrida que esporadicamente fazia com que as instâncias iniciadas fossem imediatamente encerradas porque elas ainda não estavam disponíveis na resposta do EC2. `DescribeInstances`
- Corrige o suporte para o parâmetro `DisableSimultaneousMultithreading` em tipos de

instância com processadores Arm.

- Corrija a falha na atualização da pilha de AWS ParallelCluster API ao atualizar de uma versão anterior. Adiciona o padrão de recurso usado para a Ação ListImage PipelineImages no EcrImageDeletionLambdaRole .
- Corrige a AWS ParallelCluster API adicionando as permissões ausentes necessárias para importar ou exportar do Amazon S3 ao criar um sistema de arquivos FSx for Lustre.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster atualizações somente de documentação deste ano até o momento](#)

AWS ParallelCluster atualizações somente de documentação.

6 de julho de 2022

Seções novas:

- [Práticas recomendadas: alertas de orçamento](#) V3
- [Práticas recomendadas: mover um cluster para uma nova versão de patch ou secundária do AWS ParallelCluster.](#) V3
- [Como trabalhar com o Amazon S3](#) V3
- [Trabalho com Instâncias spot](#) V3
- [Modo protegido por cluster do Slurm](#) V3
- [Recursos do AWS ParallelCluster e marcação](#) V3
- [Painel do Amazon CloudWatch](#) V3
- [Integração com Amazon CloudWatch Logs](#) V3
- [Elastic Fabric Adapter](#) V3
- [AWS ParallelCluster Personalização da AMI](#) V3
- [Iniciar instâncias com ODCR \(On-Demand Capacity Reservations\)](#) V3

- [Aplicação de patches de AMI e substituição de instâncias do EC2 V3](#)
- [Como funciona o AWS ParallelCluster V3](#)
- [Configurando a criptografia de armazenamento compartilhado com uma chave AWS KMS V3](#)
- [Executando trabalhos em um cluster de modo de várias filas V3](#)
- [Uso da API AWS ParallelCluster V3](#)

Atualizações da seção:

- [Práticas recomendadas: desempenho da rede V3](#): Foram adicionadas as melhores práticas para usar o Elastic Fabric Adaptor.
- [AWS Identity and Access Management permissões em AWS ParallelCluster V3](#): Várias atualizações e [Política de usuário adicional pcluster do AWS ParallelCluster ao usar o Amazon FSx para Lustre](#) adicionado.
- [Solução de problemas do AWS ParallelCluster V3](#): Várias atualizações.

[AWS ParallelCluster versão
3.1.4 lançada](#)

AWS ParallelCluster versão
3.1.4 lançada.

16 de maio de 2022

Melhorias:

- Adiciona validação [DirectoryService](#) / [PasswordSecretArn](#) para falha se o segredo não existir.

Adiciona suporte para habilitar a autenticação JWT do Slurm.

Alterações:

- Atualização do Slurm para a versão 21.08.8-2.
- Compilação Slurm com suporte do JWT.
- Não exige que [Placement Group](#) / [Enabled](#) seja definido como true ao passar um Placement Group / Id existente.
- Adicione `lambda:TagResource` ao `ParallelClusterUseRole` usado pela pilha de ParallelCluster APIs para criação de clusters e criação de imagens.

Correções de erros:

- Corrige a capacidade de exportar os logs de um cluster ao usar o comando `export-cluster-logs` com a opção `--filters` .
- Corrija o ponto de entrada do AWS Batch Docker para usar o diretório `/home` compartilhado para coordenar a execução de trabalhos paralelos de vários nós.
- Redefine o endereço do nó ao definir o nó estático não íntegro do slurm como inativo para evitar tratar a falha do nó estático com capacidade insuficiente como um nó de falha de bootstrap.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão 3.1.3 lançada](#)

AWS ParallelCluster versão 3.1.3 lançada.

20 de abril de 2022

Melhorias:

- Executa a criação da chave SSH junto com a criação do diretório HOME, por exemplo, durante o login SSH, ao alternar para outro usuário e ao executar um comando como outro usuário.
- Adiciona suporte para nomes distintos de FQDN e LDAP no parâmetro de configuração [Directory Service](#) / [DomainName](#). O novo validador agora verifica as duas sintaxes.
- O novo script `update_directory_service_password.sh` implantado no nó principal oferece suporte à atualização manual da senha do Active Directory na configuração do SSSD. A senha é recuperada pelo AWS Secrets Manager da configuração do cluster.
- Adiciona suporte para implantar a infraestrutura de

API em ambientes sem uma VPC padrão.

Alterações:

- Desativa C-States mais profundos em AMIs oficiais x86_64 e AMIs criadas por meio do comando `build-image`, para garantir alto desempenho e baixa latência.
- Atualizações no pacote do sistema operacional e correções de segurança.
- Altera as imagens básicas do Amazon Linux 2 para usar AMIs com o Kernel 5.10.

Correções de erros:

- Corrige a pilha de imagens de compilação em `DELETE_FAILED` após a criação bem-sucedida da imagem, devido às novas políticas do EC2 Image Builder.
- Corrige a conversão do parâmetro de configuração `DirectoryService` / `DomainAddr` para a propriedade `SSSD ldap_uri` quando ela contém vários endereços de domínio.

Para obter detalhes sobre as mudanças, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster versão 3.1.2 lançada](#)

AWS ParallelCluster versão 3.1.2 lançada.

2 de março de 2022

Alterações:

- Atualização do Slurm para a versão 21.08.6 (era 21.08.5)

Correções de erros:

- Corrige a atualização do arquivo `/etc/hosts` nos nós de computação quando um cluster é implantado em sub-redes sem acesso à Internet.
- Corrige o bootstrap dos nós de computação para aguardar a inicialização das unidades efêmeras antes de ingressar no cluster.

Para obter detalhes sobre as mudanças, consulte os CHANGELOG arquivos do pacote [aws-parallelcluster](#) em. GitHub

[AWS ParallelCluster versão 3.1.1 lançada](#)

AWS ParallelCluster versão 3.1.1 lançada.

10 de fevereiro de 2022

- Adiciona suporte para ambientes de cluster de vários usuários por meio da [integração com domínios do Active Directory \(AD\)](#) gerenciados por meio do AWS Directory Service.
- Adiciona suporte para [UseEc2Hostnames](#) no arquivo de configuração do cluster. Quando definido como verdadeiro, usa nomes de host padrão do EC2 (por exemplo, ip-1-2-3-4) para nós de computação.
- Adiciona suporte para criação de clusters em [sub-redes sem acesso à Internet](#).
- Adiciona suporte para vários tipos de instância de computação por fila.
- Adiciona suporte para agendamento de GPU com o Slurm em instâncias ARM com placas NVIDIA.
- Adicione sinalizadores abreviados para `cluster-name (-n)`, `()`, `region (-r)`, `ecluster-configuration /image-idimage-`

`configuration (-i) -c`
à CLI. AWS ParallelCluster

- Adiciona suporte para a opção `NEW_CHANGE_DELETED` para o parâmetro [AutoImportPolicy](#) do FSx para Lustre.
- Adicione uma tag `parallelcluster:compute-resource-name` aos recursos `LaunchTemplates` do EC2 usados pelos nós de computação.
- Melhora os grupos de segurança criados dentro do cluster para permitir conexões de entrada de grupos de segurança personalizados quando os parâmetros `SecurityGroups` forem especificados para alguns nós principais e/ou filas.
- Instala os drivers NVIDIA e a biblioteca CUDA para ARM.

Alterações:

- Atualização do Slurm para a versão 21.08.5 (era 20.11.8)

- Atualização do plugin do Slurm para a versão 21.08 (era 20.11)
- Atualização do NICE DCV para a versão 2021.3-11591 (era 2021.1-10851).
- Atualização do driver NVIDIA para a versão 470.103.01 (era 470.57.02).
- Atualização do NVIDIA Fabric Manager para a versão 470.103.01 (era 470.57.02).
- Atualização da biblioteca CUDA para a versão 11.4.4 (era 11.4.0).
- [Intel MPI](#) atualizado para a versão 2021 atualização 4 (atualizado a partir da versão 2019 atualização 8). Para obter mais informações, consulte [Intel® MPI Library 2021 Update 4](#).
- Atualização da biblioteca PMIx para a versão 3.2.3 (era 3.1.5).
- Remove o despejo de nós de computação com falha em /home/logs/compute. Os arquivos de log dos nós de computação estão disponíveis nos

registros do console EC2
CloudWatch e nos mesmos.

- Possibilita a supressão de validadores de comprimento `SlurmQueues` e `ComputeResources`.
- Desabilita a atualização do pacote no momento do lançamento da instância no Amazon Linux 2.
- Desative os metadados de imagem `ImageBuilder` aprimorados do EC2 ao criar imagens AWS ParallelCluster personalizadas.
- Define explicitamente a fonte de dados `cloud-init` como EC2. Isso economiza tempo de inicialização para as plataformas Ubuntu e CentOS.
- Usa o nome do recurso de computação em vez do tipo de instância no nome do modelo de lançamento da frota de computação.
- Redireciona `stderr` e `stdout` para o arquivo de log da CLI para evitar texto indesejado na saída da CLI do `pcluster`.
- Move as receitas de configuração/instalação para livros de receitas

separados que são chamados a partir do principal. Os pontos de entrada existentes são mantidos e compatíveis com versões anteriores.

- Faça o download das dependências da plataforma Intel HPC durante o tempo de construção da AMI para evitar o contato com a Internet durante a criação do cluster.
- Não use o nome - do recurso de computação ao configurar os nós do Slurm.
- Não configure GPUs no Slurm quando o driver NVIDIA não estiver instalado.
- Corrija a permissão `ecs:ListContainerInstances` em `BatchUserRole` .
- Corrija a exportação de registros de cluster quando não há prefixo especificado, anteriormente exportado para um prefixo `None`.
- Corrija a reversão que não está sendo executada em caso de falha na atualização do cluster.
- Corrija a permissão `ecs:ListContainerI`

instances em BatchUser
Role .

- Corrige o esquema RootVolume para o HeadNode gerando um erro se um KmsKeyId não suportado for especificado.
- Corrija as métricas ausentes do Amazon FSx a serem exibidas no CloudWatch painel.
- Corrige EfaSecurityGroupValidator . Anteriormente, ele tinha o potencial de produzir falsas falhas quando grupos de segurança personalizados eram fornecidos e o EFA era ativado.

Para obter detalhes sobre as alterações, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão
3.0.3 lançada](#)

AWS ParallelCluster versão
3.0.3 lançada.

17 de janeiro de 2022

- Desativa o atendente `log4j-cve-2021-442` `28-hotpatch` (`Log4jHotPatch`) no Amazon Linux 2 para evitar uma possível degradação do desempenho. Para obter mais informações, consulte o [anúncio do Amazon Linux Hotpatch para o Apache Log4j](#).

Para obter detalhes sobre as mudanças, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster versão 3.0.2 lançada](#)

AWS ParallelCluster versão 3.0.2 lançada.

5 de novembro de 2021

Atualização do instalador [Elastic Fabric Adapter](#) para 1.14.1

- Config. EFA: efa-confi g-1.9-1 (era efa-confi g-1.9)
- Perfil EFA: efa-profi le-1.5-1 (era efa-profile-1.5)
- Módulo Kernel EFA: efa-1.14.2 (era efa-1.13.0)
- Núcleo RDMA: rdma-core -37.0 (de rdma-core -35)
- Libfabric: libfabric -1.13.2 (de libfabric -1.13.0)
- Open MPI: openmpi40 -aws-4.1.1-2 (sem alteração)

O GPUDirect RDMA fica sempre ativado se for compatível com o tipo de instância. A opção [GdrSupport](#) de configuração não tem efeito.

Para obter detalhes sobre as mudanças, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão 3.0.1 lançada](#)

AWS ParallelCluster versão 3.0.1 lançada.

27 de outubro de 2021

Ferramenta de migração de configuração de cluster

- Agora, os clientes podem migrar suas configurações de cluster do formato da AWS ParallelCluster versão 2 para o formato da versão 3 baseado em YAML AWS ParallelCluster . Para ter mais informações, consulte [pcluster3-config-converter](#) .

O nó principal pode ser interrompido

- [Depois de interromper a frota computacional, o nó principal pode ser interrompido e depois reiniciado usando o console do Amazon EC2 ou o comando `stop-instances`. AWS CLI](#)

Região da AWS Leitura padrão do `~/.aws/config` arquivo

- Para o [pcluster](#) comando, se o não Região da AWS for especificado no arquivo

de configuração, no ambiente ou na linha de comando, o padrão Região da AWS especificado na `region` configuração na `[default]` seção do `~/.aws/config` arquivo será usado.

Para obter detalhes sobre as mudanças, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbook](#)[aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versão 3.0.0 lançada](#)

AWS ParallelCluster versão 3.0.0 lançada.

10 de setembro de 2021

Suporte para gerenciamento de clusters via Amazon API Gateway

- Agora, os clientes podem gerenciar e implantar clusters por meio de endpoints HTTP com o Amazon API Gateway. Isso abre novas possibilidades para fluxos de trabalho com scripts ou orientados por eventos.

A interface de linha de AWS ParallelCluster comando (CLI) também foi redesenhada para ser compatível com essa API e inclui uma nova opção de saída JSON. Essa nova funcionalidade possibilita que os clientes também implementem recursos de blocos de construção semelhantes usando a CLI.

Melhoria da criação de AMI personalizada

- Agora, os clientes têm acesso a um processo mais robusto para criar e

gerenciar AMIs personalizadas usando o EC2 Image Builder. Agora, as AMIs personalizadas podem ser gerenciadas por meio de um arquivo de AWS ParallelCluster configuração separado e podem ser criadas usando o [pcluster build-image](#) comando na interface da linha de AWS ParallelCluster comando.

Para obter detalhes sobre as mudanças, consulte os CHANGELOG arquivos do [aws-parallelcluster](#) e os pacotes em. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.