Manual do usuário

AWS PCS

AWS PCS: Manual do usuário

Table of Contents

O que é serviço de computação AWS paralela?

AWS O Serviço de Computação Paralela (AWS PCS) é um serviço gerenciado que facilita a execução e a escalabilidade de cargas de trabalho de computação de alto desempenho (HPC) e a criação de modelos científicos e de engenharia AWS usando o Slurm. Use o AWS PCS para criar clusters de computação que integram a melhor AWS computação, armazenamento, rede e visualização da categoria. Execute simulações ou crie modelos científicos e de engenharia. Simplifique e simplifique suas operações de cluster usando recursos integrados de gerenciamento e observabilidade. Capacite seus usuários a se concentrarem em pesquisa e inovação, permitindo que eles executem seus aplicativos e trabalhos em um ambiente familiar.

Tópicos

Conceitos em AWS PCS

Um cluster no AWS PCS tem 1 ou mais filas, associadas a pelo menos 1 grupo de nós de computação. Os trabalhos são enviados para filas e executados em EC2 instâncias definidas por grupos de nós de computação. Você pode usar essas bases para implementar arquiteturas de HPC sofisticadas.

Cluster

Um cluster é um recurso para gerenciar recursos e executar cargas de trabalho. Um cluster é um recurso AWS PCS que define um conjunto de configurações de computação, rede, armazenamento, identidade e agendador de tarefas. Você cria um cluster especificando qual agendador de trabalhos deseja usar (Slurm atualmente), qual configuração de agendador deseja, qual controlador de serviço deseja gerenciar o cluster e em qual VPC você deseja que os recursos do cluster sejam lançados. O agendador aceita e agenda trabalhos e também inicia os nós de computação (EC2 instâncias) que processam esses trabalhos.

Grupo de nós de computação

Um grupo de nós de computação é uma coleção de nós de computação que o AWS PCS usa para executar trabalhos ou fornecer acesso interativo a um cluster. Ao definir um grupo de nós de computação, você especifica características comuns, como tipos de EC2 instância da Amazon, contagem mínima e máxima de instâncias, sub-redes VPC de destino, Amazon Machine Image

(AMI), opção de compra e configuração de lançamento personalizada. AWS O PCS usa essas configurações para iniciar, gerenciar e encerrar com eficiência os nós de computação em um grupo de nós de computação.

Fila

Quando quiser executar um trabalho em um cluster específico, você o envia para uma fila específica (também chamada de partição). O trabalho permanece na fila até que o AWS PCS o programe para execução em um grupo de nós de computação. Você associa um ou mais grupos de nós de computação a cada fila. É necessária uma fila para agendar e executar trabalhos nos recursos do grupo de nós de computação subjacentes usando várias políticas de agendamento oferecidas pelo agendador de trabalhos. Os usuários não enviam trabalhos diretamente para um nó de computação ou grupo de nós de computação.

Administrador de sistema

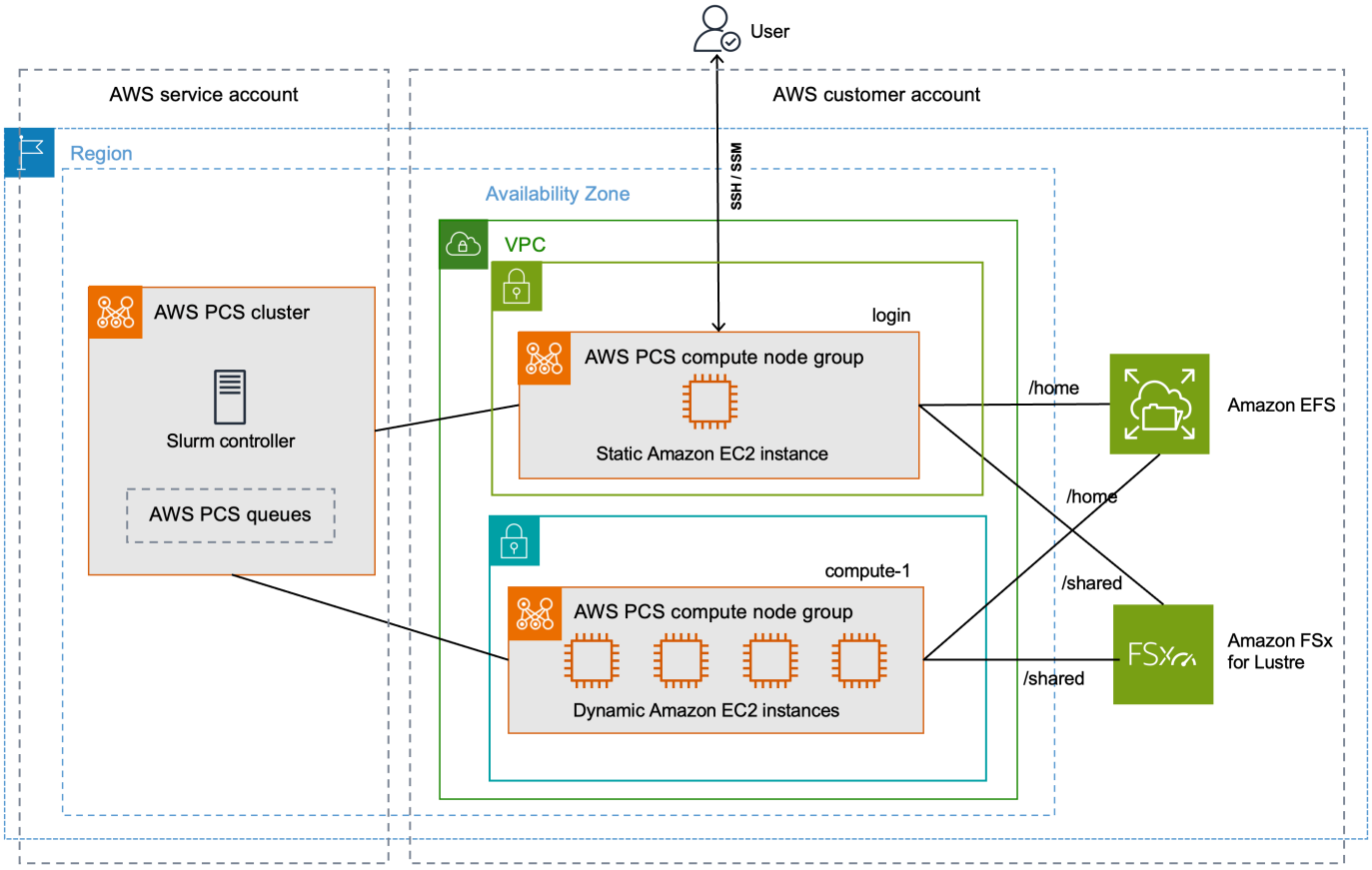
Um administrador do sistema implanta, mantém e opera um cluster. Eles podem acessar o AWS PCS por meio da Console de gerenciamento da AWS API AWS PCS e do AWS SDK. Eles têm acesso a clusters específicos por meio de SSH ou AWS Systems Manager, onde podem executar tarefas administrativas, executar trabalhos, gerenciar dados e realizar outras atividades baseadas em shell. Para obter mais informações, consulte a Documentação do AWS Systems Manager.

Usuário final

Um usuário final não tem a day-to-day responsabilidade de implantar ou operar um cluster. Eles usam uma interface de terminal (como SSH) para acessar recursos do cluster, executar trabalhos, gerenciar dados e realizar outras atividades baseadas em shell.

Comece a usar o serviço de computação AWS paralela

Este é um tutorial para criar um cluster simples que você pode usar para testar o AWS PCS. A figura a seguir mostra o design do cluster.



O tutorial de design de cluster tem os seguintes componentes principais:

- Uma VPC e sub-redes que atendem aos requisitos AWS de rede do PCS.
- Um sistema de arquivos Amazon EFS, que será usado como um diretório inicial compartilhado.
- Um sistema de arquivos Amazon FSx for Lustre, que fornece um diretório compartilhado de alto desempenho.
- Um cluster AWS PCS, que fornece um controlador Slurm.
- 2 grupos de nós de computação AWS PCS.
  - O grupo de `login` nós, que fornece acesso interativo baseado em shell ao sistema.
  - O grupo de `compute-1` nós fornece instâncias com escalabilidade elástica para executar trabalhos.

- 1 fila que envia trabalhos para EC2 instâncias no grupo de `compute-1` nós.

O cluster exige AWS recursos adicionais, como grupos de segurança, funções do IAM e modelos de EC2 execução, que não são mostrados no diagrama.

> **Note**
>
> Recomendamos que você conclua as etapas da linha de comando neste tópico em um shell do Bash. Se não estiver utilizando um shell Bash, alguns comandos de script, como caracteres de continuação de linha e a forma como as variáveis são definidas e utilizadas, exigirão o ajuste do seu shell. Além disso, as regras de citação e de escape do seu shell podem ser diferentes. Para obter mais informações, consulte Aspas e literais com cadeias de caracteres no Guia do AWS CLIAWS Command Line Interface usuário da versão 2.

Tópicos

Pré-requisitos para começar a usar o PCS AWS

Consulte os tópicos a seguir para preparar seu ambiente Conta da AWS de desenvolvimento local para o AWS PCS.

Tópicos

Inscreva-se AWS e crie um usuário administrativo

Conclua as tarefas a seguir para configurar o Serviço de Computação AWS Paralela (AWS PCS).

Tópicos

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

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

Proteja seu Usuário raiz da conta da AWS

1. Faça login Console de gerenciamento da AWScomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a 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. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o Centro de Identidade do AWS IAM no Guia do usuário do Centro de Identidade do AWS IAM .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   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 Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do Centro de Identidade do AWS IAM .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do Centro de Identidade do AWS IAM .

Instale o AWS CLI para AWS PCS

Você deve usar a versão mais recente do AWS CLI. Para obter informações, consulte Instalar ou atualizar para a versão mais recente do AWS CLI no Guia AWS Command Line Interface do Usuário da Versão 2.

Você deve configurar AWS CLI o. Para obter mais informações, consulte Configurar o AWS CLI no Guia AWS Command Line Interface do usuário para a versão 2.

Digite o seguinte comando em um prompt de comando para verificar seu AWS CLI; ele deve exibir informações de ajuda.

```
aws pcs help
```

Permissões do IAM necessárias para AWS PCS

O diretor de segurança do IAM que você está usando deve ter permissões para trabalhar com funções do IAM do AWS PCS, funções vinculadas ao serviço AWS CloudFormation, uma VPC e recursos relacionados. Para obter mais informaçõesIdentity and Access Management for AWS Parallel Computing Service, consulte e Criar uma função vinculada ao serviço no Guia do AWS Identity and Access Management usuário. Você deve concluir todas as etapas deste manual como o mesmo usuário. Execute o seguinte comando para verificar o usuário atual:

```
aws sts get-caller-identity
```

Usando AWS CloudFormation com o tutorial AWS PCS

O tutorial do AWS PCS tem várias etapas e tem como objetivo ajudá-lo a entender as partes de um cluster AWS PCS e os procedimentos necessários para criá-lo. Recomendamos que você siga as etapas do tutorial pelo menos uma vez. Depois de ter uma boa compreensão do que está envolvido, você pode usar AWS CloudFormation para criar rapidamente o cluster de amostra com automação.

CloudFormation é um AWS serviço que permite criar e provisionar implantações de AWS infraestrutura de forma previsível e repetida. Você pode usar um CloudFormation modelo para provisionar automaticamente os AWS recursos para o cluster de amostra como uma única unidade, chamada de pilha. Você pode excluir a pilha quando terminar de usá-la.

Para obter mais informações, consulte Comece a usar um CloudFormation AWS PCS.

Crie uma VPC e sub-redes para PCS AWS

Você pode criar uma VPC e sub-redes com um modelo. CloudFormation Use o URL a seguir para baixar o CloudFormation modelo e, em seguida, faça o upload do modelo no CloudFormation console para criar uma nova CloudFormation pilha. Para obter mais informações, consulte Usando o CloudFormation console no Guia AWS CloudFormation do usuário.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/net/hpc_large_scale/assets/
main.yaml
```

Com o modelo aberto no CloudFormation console, insira as seguintes opções. Você pode usar os valores padrão fornecidos no modelo.

- Em Forneça um nome de pilha:
  - Em Nome da pilha, digite:

    ```
    hpc-networking
    ```
- Em Parâmetros:
  - Em VPC:

    - Em CidrBlock, insira:

      ```
      10.3.0.0/16
      ```

  - Em Sub-redes A:
    - Em CidrPublicSubnetA, digite:

      ```
      10.3.0.0/20
      ```

    - Em CidrPrivateSubnetA, digite:

      ```
      10.3.128.0/20
      ```

  - Em Sub-redes B:
    - Em CidrPublicSubnetB, insira:

      ```
      10.3.16.0/20
      ```

    - Em CidrPrivateSubnetB, insira:

      ```
      10.3.144.0/20
      ```

  - Em Sub-redes C:
    - Para ProvisionSubnetsC, selecione True
    - Em CidrPublicSubnetC, insira:

      ```
      10.3.32.0/20
      ```

    - Em CidrPrivateSubnetC, insira:

      ```
      10.3.160.0/20
      ```

- Em Capacidades:
  - Marque a caixa “Eu reconheço que isso AWS CloudFormation pode criar recursos do IAM”.

Monitore o status da CloudFormation pilha. Quando chegar`CREATE_COMPLETE`, encontre o ID do grupo de segurança padrão na nova VPC. Você usa o ID posteriormente no tutorial.

Encontre o grupo de segurança padrão para o cluster VPC

Para encontrar o ID do grupo de segurança padrão na nova VPC, siga este procedimento:

- Navegue até o console da Amazon VPC.
- No painel da VPC, selecione Filtrar por VPC.
  - Escolha a VPC com a qual o nome começa. `hpc-networking`
  - Em Segurança, escolha Grupos de segurança.
- Encontre o ID do grupo de segurança para o grupo chamado`default`. Tem a descrição`default VPC security group`. Posteriormente, você usa o ID para configurar os modelos de execução do EC2.

Crie grupos de segurança para AWS PCS

AWS O PCS depende de grupos de segurança para gerenciar o tráfego de rede que entra e sai de um cluster e seus grupos de nós de computação. Para obter informações detalhadas sobre esse tópico, consulteRequisitos e considerações do grupo de segurança.

Nesta etapa, você usará um CloudFormation modelo para criar dois grupos de segurança.

- Um grupo de segurança de cluster, que permite a comunicação entre o controlador AWS PCS, os nós de computação e os nós de login.
- Um grupo de segurança SSH de entrada, que você pode adicionar opcionalmente aos seus nós de login para oferecer suporte ao acesso SSH

Crie os grupos de segurança para AWS PCS

Você pode usar um CloudFormation modelo para criar os grupos de segurança. Use o URL a seguir para baixar o CloudFormation modelo e, em seguida, faça o upload do modelo no CloudFormation console para criar uma nova CloudFormation pilha. Para obter mais informações, consulte Usando o CloudFormation console no Guia AWS CloudFormation do usuário.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-cluster-sg.yaml
```

Com o modelo aberto no AWS CloudFormation console, insira as seguintes opções. Observe que algumas opções serão pré-preenchidas no modelo — você pode simplesmente deixá-las como valores padrão.

- Em Forneça um nome de pilha
  - Em Nome da pilha, digite:

    ```
    getstarted-sg
    ```

- Em Parâmetros
  - Em VpcId, escolha a VPC com a qual o nome começa. `hpc-networking`
  - (Opcional) Em ClientIpCidr, insira um intervalo de IP mais restritivo para o grupo de segurança SSH de entrada. Recomendamos que você restrinja isso com seu próprio IP/sub-rede (x.x.x.x/32 para seu próprio ip ou x.x.x.x/24 para intervalo. Substitua x.x.x.x pelo seu próprio IP PÚBLICO. Você pode obter seu IP público usando ferramentas como https://ifconfig.co/)

Monitore o status da CloudFormation pilha. Quando chega ao grupo `CREATE_COMPLETE` de segurança, os recursos estão prontos.

Há dois grupos de segurança criados, com os nomes:

- `cluster-getstarted-sg`— este é o grupo de segurança do cluster
- `inbound-ssh-getstarted-sg`— este é um grupo de segurança para permitir acesso SSH de entrada

Crie um cluster no AWS PCS

No AWS PCS, um cluster é um recurso persistente para gerenciar recursos e executar cargas de trabalho. Você cria um cluster para um agendador específico (o AWS PCS atualmente oferece suporte ao Slurm) em uma sub-rede de uma VPC nova ou existente. O cluster aceita e agenda trabalhos e também inicia os nós de computação (EC2 instâncias) que processam esses trabalhos.

Para criar um cluster

1. Abra o console AWS PCS e escolha Criar cluster.
2. Na seção Detalhes do cluster, insira os seguintes campos:

- Nome do cluster — Enter `get-started`
- Scheduler — Selecione a versão 25.05 do Slurm
- Tamanho do controlador — Selecione Pequeno

3. Na seção Rede, selecione valores para os seguintes campos:
   - VPC — Escolha a VPC chamada `hpc-networking:Large-Scale-HPC`
   - Sub-rede — Selecione a sub-rede em que o nome começa com `hpc-networking:PrivateSubnetA`
   - Grupos de segurança — Selecione o grupo de segurança do cluster chamado `cluster-getstarted-sg`
4. Selecione Criar cluster.

> **Note**
>
> O campo Status mostra Como criar enquanto o cluster está sendo provisionado. A criação do cluster pode levar vários minutos.

Crie armazenamento compartilhado para AWS PCS no Amazon Elastic File System

O Amazon Elastic File System (Amazon EFS) é um AWS serviço que fornece armazenamento de arquivos totalmente elástico e sem servidor para que você possa compartilhar dados de arquivos sem provisionar ou gerenciar a capacidade e o desempenho do armazenamento. Para obter mais informações, consulte What is Amazon Elastic File System? no Guia do usuário do Amazon Elastic File System.

O cluster de demonstração do AWS PCS usa um sistema de arquivos EFS para fornecer um diretório inicial compartilhado entre os nós do cluster. Crie um sistema de arquivos EFS na mesma VPC do seu cluster.

Como criar seu sistema de arquivos do Amazon EFS

1. Acesse o console do Amazon EFS.

2. Certifique-se de que esteja configurado da mesma forma Região da AWS em que você experimentará o AWS PCS.
3. Escolha Create file system (Criar sistema de arquivos).
4. Na página Criar sistema de arquivos, defina os seguintes parâmetros:
   - Em Nome, insira `getstarted-efs`.
   - Em Virtual Private Cloud (VPC), escolha a VPC chamada `hpc-networking:Large-Scale-HPC`
   - Escolha Criar. Isso o levará de volta à página Sistemas de arquivos.
5. Anote a ID do sistema de arquivos do sistema de `getstarted-efs` arquivos. Você usa essas informações posteriormente.

Crie armazenamento compartilhado para AWS PCS no Amazon FSx for Lustre

O Amazon FSx for Lustre torna fácil e econômico lançar e executar o popular sistema de arquivos Lustre de alto desempenho. É possível usar o Lustre para workloads em que a velocidade é importante, como machine learning, computação de alta performance (HPC), processamento de vídeo e modelagem financeira. Para obter mais informações, consulte O que é o Amazon FSx for Lustre? no Guia do usuário do Amazon FSx for Lustre.

O cluster de demonstração do AWS PCS pode usar um FSx sistema de arquivos for Lustre para fornecer um diretório compartilhado de alto desempenho entre os nós do cluster. Crie um sistema de arquivos FSx for Lustre na mesma VPC do seu cluster.

Para criar seu sistema de arquivos FSx for Lustre

1. Acesse o FSx console da Amazon.
2. Verifique se o console está configurado para usar o Região da AWS mesmo que seu cluster.
3. Escolha Create file system (Criar sistema de arquivos).
   - Em Selecionar tipo de sistema de arquivos, escolha Amazon FSx for Lustre e, em seguida, escolha Avançar.
4. Na página Especificar detalhes do sistema de arquivos, defina os seguintes parâmetros:
   - Em Detalhes do sistema de arquivos

- Em Nome, insira `getstarted-fsx`.
- Para o tipo de implantação e armazenamento, escolha Persistente, SSD
- Para taxa de transferência por unidade de armazenamento, escolha 125 MB/s/TiB
- Em Capacidade de armazenamento, insira 1,2 TiB
- Para Configuração de metadados, escolha Automático
- Para Tipo de compactação de dados, escolha LZ4

- Em Rede e segurança
  - Para Virtual Private Cloud (VPC), escolha a VPC chamada `hpc-networking:Large-Scale-HPC`
  - Para grupos de segurança da VPC, deixe o grupo de segurança chamado `default`
  - Em Sub-rede, escolha a sub-rede em que o nome começa com `hpc-networking:PrivateSubnetA`
- Deixe as outras opções definidas com seus valores padrão.
- Escolha Próximo.

5. Na página Revisar e criar, escolha Criar sistema de arquivos. Isso o levará de volta à página Sistemas de arquivos.
6. Navegue até a página de detalhes do sistema de arquivos FSx for Lustre que você criou.
7. Anote a ID do sistema de arquivos e o nome da montagem. Você usa essas informações posteriormente.

> **Note**
>
> O campo Status mostra Criando enquanto o sistema de arquivos está sendo provisionado. A criação do sistema de arquivos pode levar vários minutos. Espere até que ele seja concluído antes de continuar com o restante do tutorial.

Crie grupos de nós de computação no AWS PCS

Um grupo de nós de computação é uma coleção virtual de nós de computação (instâncias EC2) que o AWS PCS inicia e gerencia. Ao definir um grupo de nós de computação, você especifica características comuns, como tipos de instância EC2, contagem mínima e máxima de instâncias, sub-redes VPC de destino, opção de compra preferencial e configuração de execução

personalizada. AWS O PCS inicia, gerencia e encerra com eficiência os nós de computação em um grupo de nós de computação, de acordo com essas configurações. O cluster de demonstração usa um grupo de nós de computação para fornecer nós de login para acesso do usuário e um grupo de nós de computação separado para processar trabalhos. Os tópicos a seguir descrevem os procedimentos para configurar esses grupos de nós de computação em seu cluster.

Tópicos

Crie um perfil de instância para AWS PCS

Os grupos de nós de computação exigem um perfil de instância quando são criados. Se você usar o Console de gerenciamento da AWS para criar uma função para o Amazon EC2, o console criará automaticamente um perfil de instância e dará a ele o mesmo nome da função. Para obter mais informações, consulte Como usar perfis de instância no Guia AWS Identity and Access Management do usuário.

No procedimento a seguir, você usa o Console de gerenciamento da AWS para criar uma função para o Amazon EC2, que também cria o perfil de instância para seus grupos de nós de computação.

Para criar a função e o perfil da instância

- Navegue até o console do IAM.
- Em Access management (Gerenciamento de acesso), escolha Policies (Políticas).
  - Selecione Criar política.
  - Em Especificar permissões, em Editor de políticas, escolha JSON.
  - Substitua o conteúdo do editor de texto pelo seguinte:

    JSON

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

```
                "pcs:RegisterComputeNodeGroupInstance"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

- Escolha Próximo.
- Em Revisar e criar, em Nome da política, insira`AWSPCS-getstarted-policy`.
- Selecione Criar política.

- Em Access management (Gerenciamento de acesso), escolha Roles (Funções).
- Selecione Criar perfil.
- Em Selecionar entidade confiável:
  - Para Tipo de entidade confiável, selecione AWS serviço
  - Em Caso de uso, selecione EC2.
    - Em seguida, em Escolha um caso de uso para o serviço especificado, escolha EC2.
  - Escolha Próximo.
- Em Adicionar permissões:
  - Em Políticas de permissões, pesquise por AWSPCS-getstarted-policy.
  - Marque a caixa ao lado AWSPCS-getstartedde -policy para adicioná-la à função.
  - Em Políticas de permissões, pesquise por Amazon SSMManaged InstanceCore.
  - Marque a caixa ao lado da Amazon SSMManaged InstanceCore para adicioná-la à função.
  - Escolha Próximo.
- Em Nome, revise e crie:
  - Em Detalhes da função:
    - Em Nome do perfil, insira `AWSPCS-getstarted-role`.
  - Escolha Create role (Criar função).

Crie modelos de lançamento para AWS PCS

Ao criar um grupo de nós de computação, você fornece um modelo de execução do EC2 que o AWS PCS usa para configurar as instâncias do EC2 que ele executa. Isso inclui configurações como grupos de segurança e scripts que são executados quando a instância é executada.

Nesta etapa, um CloudFormation modelo será usado para criar dois modelos de lançamento do EC2. Um modelo será usado para criar nós de login e o outro será usado para criar nós de computação. A principal diferença entre eles é que os nós de login podem ser configurados para permitir acesso SSH de entrada.

Acesse o CloudFormation modelo

Use o URL a seguir para baixar o CloudFormation modelo e, em seguida, faça o upload do modelo no CloudFormation console para criar uma nova CloudFormation pilha. Para obter mais informações, consulte Usando o CloudFormation console no Guia AWS CloudFormation do usuário.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-lt-efs-fsxl.yaml
```

Use o CloudFormation modelo para criar modelos de lançamento do EC2

Use o procedimento a seguir para preencher o CloudFormation modelo no CloudFormation console

- Em Forneça um nome de pilha:
  - Em Nome da pilha, insira`getstarted-lt`.
- Em Parâmetros:
  - Em Segurança
    - Para VpcSecurityGroupId, selecione o grupo de segurança nomeado `default` em seu cluster VPC.
    - Para ClusterSecurityGroupId, selecione o grupo chamado `cluster-getstarted-sg`
    - Para SshSecurityGroupId, selecione o grupo chamado `inbound-ssh-getstarted-sg`
    - Para SshKeyName, selecione seu par de chaves SSH preferido.
  - Em Sistemas de arquivos
    - Para EfsFilesystemId, insira a ID do sistema de arquivos EFS que você criou anteriormente no tutorial.
    - Para FSxLustreFilesystemId, insira o ID do sistema de arquivos do FSx Lustre que você criou anteriormente no tutorial.
    - Para FSxLustreFilesystemMountName, insira o nome de montagem para o mesmo FSx sistema de arquivos Lustre.
- Escolha Avançar e, em seguida, escolha Avançar novamente.
- Selecione Enviar.

Monitore o status da CloudFormation pilha. Quando chega, `CREATE_COMPLETE` o modelo de lançamento está pronto para ser usado.

> **Note**
>
> Para ver todos os recursos criados pelo CloudFormation modelo, abra o CloudFormation console. Escolha a pilha `getstarted-lt` e depois a guia Resources (Recursos).

Crie um grupo de nós de computação para nós de login no AWS PCS

Um grupo de nós de computação é uma coleção virtual de nós de computação (instâncias EC2) que o AWS PCS inicia e gerencia. Ao definir um grupo de nós de computação, você especifica características comuns, como tipos de instância EC2, contagem mínima e máxima de instâncias, sub-redes VPC de destino, opção de compra preferencial e configuração de execução personalizada. AWS O PCS inicia, gerencia e encerra com eficiência os nós de computação em um grupo de nós de computação, de acordo com essas configurações.

Nesta etapa, você iniciará um grupo de nós de computação estático que fornece acesso interativo ao cluster. Você pode usar o SSH ou o Amazon EC2 Systems Manager (SSM) para fazer login nele, depois executar comandos de shell e gerenciar trabalhos do Slurm.

Para criar o grupo de nós de computação

- Abra o console AWS PCS e navegue até Clusters.
- Selecione o cluster chamado `get-started`
- Navegue até grupos de nós de computação e escolha Criar.
- Na seção Configuração do grupo de nós de computação, forneça o seguinte:
  - Nome do grupo de nós de computação — Enter`login`.
- Em Configuração de computação, insira ou selecione estes valores:
  - Modelo de lançamento do EC2 — Escolha o modelo de lançamento em que o nome está `login-getstarted-lt`
  - Perfil da instância do IAM — Escolha o perfil da instância chamado `AWSPCS-getstarted-role`
  - Sub-redes — Selecione a sub-rede com a qual o nome começa. `hpc-networking:PublicSubnetA`

  - Instâncias — Selecione`c6i.xlarge`.
  - Configuração de escalabilidade — Em Contagem mínima de instâncias, insira. 1 Em Contagem máxima de instâncias, insira**1**.
- Em Configurações adicionais, especifique o seguinte:
  - ID da AMI — Selecione uma AMI que você deseja usar, que tenha um nome no seguinte formato:

    ```
    aws-pcs-sample_ami-amzn2-platform-slurm-version
    ```

    Para obter mais informações sobre a amostra AMIs, consulteUsando amostras de Amazon Machine Images (AMIs) com AWS PCS.
- Escolha Criar grupo de nós de computação.

O campo Status mostra Criando enquanto o grupo de nós de computação está sendo provisionado. Você pode prosseguir para a próxima etapa do tutorial enquanto ele estiver em andamento.

Crie um grupo de nós de computação para executar trabalhos de computação no PCS AWS

Nesta etapa, você iniciará um grupo de nós de computação que se expande elasticamente para executar trabalhos enviados ao cluster.

Para criar o grupo de nós de computação

- Abra o console AWS PCS e navegue até Clusters.
- Selecione o cluster chamado `get-started`
- Navegue até grupos de nós de computação e escolha Criar.
- Na seção Configuração do grupo de nós de computação, forneça o seguinte:
  - Nome do grupo de nós de computação — Enter`compute-1`.
- Em Configuração de computação, insira ou selecione estes valores:
  - Modelo de lançamento do EC2 — Escolha o modelo de lançamento em que o nome está `compute-getstarted-lt`
  - Perfil da instância do IAM — Escolha o perfil da instância chamado `AWSPCS-getstarted-role`

- Sub-redes — Selecione a sub-rede com a qual o nome começa. `hpc-networking:PrivateSubnetA`
- Instâncias — Selecione`c6i.xlarge`.
- Configuração de escalabilidade — Em Contagem mínima de instâncias, insira. `0` Em Contagem máxima de instâncias, insira**4**.

- Em Configurações adicionais, especifique o seguinte:
  - ID da AMI — Selecione uma AMI que você deseja usar, que tenha um nome no seguinte formato:

    ```
    aws-pcs-sample_ami-amzn2-platform-slurm-version
    ```

    Para obter mais informações sobre a amostra AMIs, consulteUsando amostras de Amazon Machine Images (AMIs) com AWS PCS.
- Escolha Criar grupo de nós de computação.

O campo Status mostra Criando enquanto o grupo de nós de computação está sendo provisionado.

> **Important**
>
> Aguarde até que o campo Status mostre Ativo antes de prosseguir para a próxima etapa deste tutorial.

Crie uma fila para gerenciar trabalhos no AWS PCS

Você envia um trabalho para uma fila para executá-lo. O trabalho permanece na fila até que o AWS PCS o programe para execução em um grupo de nós de computação. Cada fila está associada a um ou mais grupos de nós de computação, que fornecem as EC2 instâncias necessárias para fazer o processamento.

Nesta etapa, você criará uma fila que usa o grupo de nós de computação para processar trabalhos.

Para criar uma fila

- Abra o console AWS PCS.
- Selecione o cluster chamado `get-started`.

- Navegue até grupos de nós de computação e verifique se o status do `compute-1` grupo é Ativo.

> **Important**
>
> O status do `compute-1` grupo deve ser Ativo antes de você prosseguir para a próxima etapa.

- Navegue até Filas e escolha Criar fila.
  - Na seção Configuração da fila, forneça os seguintes valores:
    - Nome da fila — Insira o seguinte: `demo`
    - Grupos de nós de computação — Selecione o grupo de nós de computação chamado. `compute-1`
- Selecione Criar fila.

O campo Status mostra Criando enquanto a fila está sendo criada.

> **Important**
>
> Aguarde até que o campo Status mostre Ativo antes de prosseguir para a próxima etapa deste tutorial.

Conecte-se ao seu cluster AWS PCS

Depois que o status do grupo de nós de `login` computação se tornar Ativo, você poderá se conectar à EC2 instância que ele criou.

Para se conectar ao nó de login

- Abra o console AWS PCS e navegue até Clusters.
- Selecione o cluster chamado `get-started`.
- Escolha grupos de nós de computação.
- Navegue até o grupo de nós de computação chamado`login`.
- Encontre o ID do grupo de nós de computação.
- Em outra janela ou guia do navegador, abra o EC2 console da Amazon.

- Selecione Instances (Instâncias).
- Pesquise EC2 instâncias com a seguinte tag. *node-group-id*Substitua pelo valor do ID do grupo de nós de computação da etapa anterior. Deve haver 1 instância.

```
aws:pcs:compute-node-group-id=node-group-id
```

- Conecte-se à EC2 instância. Você pode usar o Gerenciador de Sessões ou o SSH.

  Session Manager

  - Selecione a instância.
  - Selecione Conectar.
  - Em Conectar à instância, selecione Gerenciador de sessões.
  - Selecione Conectar.
  - Selecione Conectar. Um terminal interativo é iniciado em seu navegador.

  SSH

  - Selecione a instância.
  - Selecione Conectar.
  - Em Connect to instance, selecione Cliente SSH.
  - Siga as instruções fornecidas pelo console.

    > **Note**
    >
    > O nome de usuário da instância **`ec2-user`**não é`root`.

Explore o ambiente de cluster no AWS PCS

Depois de fazer login no cluster, você pode executar comandos shell. Por exemplo, você pode alterar usuários, trabalhar com dados em sistemas de arquivos compartilhados e interagir com o Slurm.

Alterar usuário

Se você fez login no cluster usando o Gerenciador de Sessões, você pode estar conectado como`ssm-user`. Esse é um usuário especial criado para o Gerenciador de Sessões. Mude para o usuário padrão no Amazon Linux 2 usando o comando a seguir. Você não precisará fazer isso se estiver conectado usando SSH.

```
sudo su - ec2-user
```

Trabalhe com sistemas de arquivos compartilhados

Você pode confirmar se o sistema de arquivos EFS e FSx os sistemas de arquivos Lustre estão disponíveis com o comando. `df -h` A saída em seu cluster deve ser semelhante à seguinte:

```
[ec2-user@ip-10-3-6-103 ~]$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   3.8G     0  3.8G   0% /dev
tmpfs                      3.9G     0  3.9G   0% /dev/shm
tmpfs                      3.9G  556K  3.9G   1% /run
tmpfs                      3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/nvme0n1p1              24G   18G  6.6G  73% /
127.0.0.1:/                8.0E     0  8.0E   0% /home
10.3.132.79@tcp:/zlshxbev  1.2T  7.5M  1.2T   1% /shared
tmpfs                      780M     0  780M   0% /run/user/0
tmpfs                      780M     0  780M   0% /run/user/1000
```

O `/home` sistema de arquivos monta 127.0.0.1 e tem uma capacidade muito grande. Esse é o sistema de arquivos EFS que você criou anteriormente no tutorial. Todos os arquivos gravados aqui estarão disponíveis `/home` em todos os nós do cluster.

O `/shared` sistema de arquivos monta um IP privado e tem uma capacidade de 1,2 TB. Esse é o sistema FSx de arquivos do Lustre que você criou anteriormente no tutorial. Todos os arquivos gravados aqui estarão disponíveis `/shared` em todos os nós do cluster.

Interaja com o Slurm

Tópicos

- Listar filas e nós
- Mostrar empregos

Listar filas e nós

Você pode listar as filas e os nós aos quais elas estão associadas ao uso`sinfo`. A saída do seu cluster deve ser semelhante à seguinte:

```
[ec2-user@ip-10-3-6-103 ~]$ sinfo
```

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
demo         up   infinite      4  idle~ compute-1-[1-4]
[ec2-user@ip-10-3-6-103 ~]$
```

Observe a partição chamadademo. Seu status é up e tem no máximo 4 nós. Está associado aos nós do grupo de compute-1 nós. Se você editar o grupo de nós de computação e aumentar o número máximo de instâncias para 8, o número de nós será lido 8 e a lista de nós será lidacompute-1-[1-8]. Se você criasse um segundo grupo de nós de computação chamado test com 4 nós e o adicionasse à demo fila, esses nós também apareceriam na lista de nós.

Mostrar empregos

Você pode listar todos os trabalhos, em qualquer estado, no sistema comsqueue. A saída do seu cluster deve ser semelhante à seguinte:

```
[ec2-user@ip-10-3-6-103 ~]$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Tente executar squeue novamente mais tarde, quando você tiver um trabalho do Slurm pendente ou em execução.

Execute uma tarefa de nó único no AWS PCS

Para executar um trabalho usando o Slurm, você prepara um script de envio especificando os requisitos do trabalho e o envia para uma fila com o comando. sbatch Normalmente, isso é feito em um diretório compartilhado para que os nós de login e computação tenham um espaço comum para acessar arquivos.

Conecte-se ao nó de login do seu cluster e execute os comandos a seguir em seu prompt de shell.

- Torne-se o usuário padrão. Mude para o diretório compartilhado.

  ```
  sudo su - ec2-user
  cd /shared
  ```

- Use os comandos a seguir para criar um exemplo de script de trabalho:

  ```
  cat << EOF > job.sh
  #!/bin/bash
  ```

```
#SBATCH -J single
#SBATCH -o single.%j.out
#SBATCH -e single.%j.err

echo "This is job \${SLURM_JOB_NAME} [\${SLURM_JOB_ID}] running on \
${SLURMD_NODENAME}, submitted from \${SLURM_SUBMIT_HOST}" && sleep 60 && echo "Job
 complete"
EOF
```

- Envie o script do trabalho para o agendador do Slurm:

```
sbatch -p demo job.sh
```

- Quando o trabalho for enviado, ele retornará uma ID do trabalho como um número. Use esse ID para verificar o status do trabalho. *job-id*Substitua o comando a seguir pelo número retornado de`sbatch`.

```
squeue --job job-id
```

Example

```
squeue --job 1
```

O `squeue` comando retorna uma saída semelhante à seguinte:

```
JOBID PARTITION NAME USER     ST TIME NODES NODELIST(REASON)
1     demo      test ec2-user CF 0:47 1     compute-1
```

- Continue verificando o status da tarefa até que ela atinja o status R (em execução). O trabalho é feito quando `squeue` não devolve nada.
- Inspecione o conteúdo do `/shared` diretório.

```
ls -alth /shared
```

A saída do comando é semelhante à seguinte:

```
-rw-rw-r- 1 ec2-user ec2-user 107 Mar 19 18:33 single.1.out
-rw-rw-r- 1 ec2-user ec2-user 0 Mar 19 18:32 single.1.err
-rw-rw-r- 1 ec2-user ec2-user 381 Mar 19 18:29 job.sh
```

Os arquivos `single.1.err` foram nomeados `single.1.out` e gravados por um dos nós de computação do seu cluster. Como o trabalho foi executado em um diretório compartilhado (`/shared`), eles também estão disponíveis em seu nó de login. É por isso que você configurou um sistema de arquivos FSx for Lustre para esse cluster.

- Inspecione o conteúdo do `single.1.out` arquivo.

```
cat /shared/single.1.out
```

A saída é semelhante à seguinte:

```
This is job test [1] running on compute-1, submitted from ip-10-3-13-181
Job complete
```

Execute uma tarefa MPI de vários nós com o Slurm no PCS AWS

Essas instruções demonstram o uso do Slurm para executar uma tarefa de interface de passagem de mensagens (MPI) no PCS. AWS

Execute os comandos a seguir em um prompt de shell do seu nó de login.

- Torne-se o usuário padrão. Mude para seu diretório inicial.

```
sudo su - ec2-user
cd ~/
```

- Crie o código-fonte na linguagem de programação C.

```
cat > hello.c << EOF
// * mpi-hello-world - https://www.mpitutorial.com
// Released under MIT License
//
// Copyright (c) 2014 MPI Tutorial.
//
// 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, subject to the following conditions:
```

```
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// 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.

#include <mpi.h>
#include <stdio.h>
#include <stddef.h>

int main(int argc, char** argv) {
  // Initialize the MPI environment. The two arguments to MPI Init are not
  // currently used by MPI implementations, but are there in case future
  // implementations might need the arguments.
  MPI_Init(NULL, NULL);

  // Get the number of processes
  int world_size;
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);

  // Get the rank of the process
  int world_rank;
  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

  // Get the name of the processor
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int name_len;
  MPI_Get_processor_name(processor_name, &name_len);

  // Print off a hello world message
  printf("Hello world from processor %s, rank %d out of %d processors\n",
         processor_name, world_rank, world_size);

  // Finalize the MPI environment. No more MPI calls can be made after this
  MPI_Finalize();
}
EOF
```

- Carregue o módulo openMPI.

```
module load openmpi
```

- Compile o programa C.

```
mpicc -o hello hello.c
```

- Escreva um script de envio de trabalhos no Slurm.

```
cat > hello.sh << EOF
#!/bin/bash
#SBATCH -J multi
#SBATCH -o multi.out
#SBATCH -e multi.err
#SBATCH --exclusive
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1

srun $HOME/hello
EOF
```

- Mude para o diretório compartilhado.

```
cd /shared
```

- Envie o roteiro do trabalho.

```
sbatch -p demo ~/hello.sh
```

- Use `squeue` para monitorar o trabalho até que seja concluído.
- Confira o conteúdo de`multi.out`:

```
cat multi.out
```

A saída é semelhante à seguinte. Observe que cada classificação tem seu próprio endereço IP porque foi executada em um nó diferente.

```
Hello world from processor ip-10-3-133-204, rank 0 out of 4 processors
Hello world from processor ip-10-3-128-219, rank 2 out of 4 processors
Hello world from processor ip-10-3-141-26, rank 3 out of 4 processors
```

```
Hello world from processor ip-10-3-143-52, rank 1 out of 4 processor
```

Exclua seus AWS recursos para AWS PCS

Depois de concluir os grupos de clusters e nós que você criou para este tutorial, você deve excluir os recursos que você criou.

> **Important**
>
> Você recebe cobranças de cobrança por todos os recursos em execução no seu Conta da AWS

Para excluir recursos do AWS PCS que você criou para este tutorial

- Abra o console AWS PCS.
- Navegue até o cluster chamado get-started.
- Navegue até a seção Filas.
- Selecione a fila chamada demo.
- Escolha Excluir.

> **Important**
>
> Espere até que a fila seja excluída antes de continuar.

- Navegue até a seção Grupos de nós de computação.
- Selecione o grupo de nós de computação chamado compute-1.
- Escolha Excluir.
- Selecione o grupo de nós de computação chamado login.
- Escolha Excluir.

> **Important**
>
> Espere até que os dois grupos de nós de computação tenham sido excluídos antes de continuar.

- Na página de detalhes do cluster para começar, escolha Excluir.

> **Important**
>
> Espere até que o cluster seja excluído antes de prosseguir com as etapas subsequentes.

Para excluir outros AWS recursos que você criou para este tutorial

- Abra o console do IAM.
  - Escolha Perfis.
  - Selecione a função chamada AWSPCS-getstarted-role e escolha Excluir.
  - Depois que a função for excluída, escolha Políticas.
  - Selecione a política chamada AWSPCS-getstarted-policy e escolha Excluir.
- Abra o console de CloudFormation.
  - Selecione a pilha chamada getstarted-lt.
  - Escolha Excluir.

> **Important**
>
> Aguarde até que a pilha seja excluída antes de continuar.

- Abra o Console do Amazon EFS .
  - Escolha Sistemas de arquivos.
  - Selecione o sistema de arquivos chamado getstarted-efs.
  - Escolha Excluir.

> **Important**
>
> Aguarde até que o sistema de arquivos seja excluído antes de continuar.

- Abra o FSx console da Amazon.
  - Escolha Sistemas de arquivos.
  - Selecione o sistema de arquivos chamado getstarted-fsx.
  - Escolha Excluir.

> **Important**
>
> Aguarde até que o sistema de arquivos seja excluído antes de continuar.

- Abra o console de CloudFormation.
  - Selecione a pilha chamada getstarted-sg.
  - Escolha Excluir.
- Abra o console de CloudFormation.
  - Selecione a pilha chamada hpc-networking.
  - Escolha Excluir.

Comece a usar um CloudFormation AWS PCS

Você pode usar AWS CloudFormation para criar um cluster AWS PCS. CloudFormation permite criar e provisionar implantações de AWS infraestrutura de forma previsível e repetida. Você pode usar CloudFormation para provisionar automaticamente recursos de vários AWS serviços para criar aplicativos altamente confiáveis, escaláveis e econômicos Nuvem AWS sem criar e configurar a infraestrutura subjacente. AWS CloudFormation permite que você use um arquivo de modelo para criar e excluir uma coleção de recursos juntos como uma única unidade, chamada de pilha. Para obter mais informações sobre CloudFormation, consulte O que é CloudFormation? no Guia do AWS CloudFormation usuário. Para obter mais informações sobre os tipos de recursos AWS PCS em CloudFormation, consulte a referência do tipo de recurso AWS PCS no Guia AWS CloudFormation do usuário.

Tópicos

Use CloudFormation para criar um cluster AWS PCS de amostra

O procedimento a seguir usa um CloudFormation modelo no Console de gerenciamento da AWS para criar um cluster AWS PCS de amostra. Para obter mais informações sobre CloudFormation, consulte O que é CloudFormation? no Guia do AWS CloudFormation usuário. Para obter mais informações sobre os tipos de recursos AWS PCS em CloudFormation, consulte a referência do tipo de recurso AWS PCS no Guia AWS CloudFormation do usuário.

Para criar o cluster de amostra

1. Escolha o Região da AWS para criar o cluster (o link abre o CloudFormation console com o modelo):

   - Leste dos EUA (Norte da Virgínia) (us-east-1)
   - Leste dos EUA (Ohio) (us-east-2)

- Oeste dos EUA (Oregon) (us-west-2)
- Ásia-Pacífico (Cingapura) (ap-southeast-1)
- Ásia-Pacífico (Sydney) (ap-southeast-2)
- Ásia-Pacífico (Tóquio) (ap-northeast-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) (eu-west-2)
- Europa (Estocolmo) (eu-north-1)
- AWS GovCloud (Leste dos EUA) (us-gov-east-1)
- AWS GovCloud (Oeste dos EUA) (us-gov-west-1)

2. Em Forneça um nome de pilha, insira um nome descritivo. Esse é o nome da sua CloudFormation pilha. O modelo usa esse valor como o nome do seu cluster AWS PCS.
3. Em Parâmetros:
   a. Em SlurmVersion, escolha a versão do Slurm que você deseja que seu cluster use.
   b. Em NodeArchitecture, escolha x86 para implantar um cluster que usa instâncias compatíveis com x86_64 ou escolha Graviton para usar instâncias Arm64.
   c. Para KeyName, escolha um par de chaves SSH para acessar os nós de login do cluster. Verifique se você tem o arquivo PEM do par de chaves escolhido.
   d. Para ClientIpCidr, insira um intervalo de IP no formato CIDR para controlar o acesso aos nós de login.

      > **Warning**
      >
      > O valor padrão de `0.0.0.0/0` permite o acesso de todos os endereços IP.

   e. Deixe os valores para o HpcRecipesS3Bucket e HpcRecipesBranchcomo seus valores padrão.
4. Em Capacidades e transformações:
   a. Marque a caixa de seleção para confirmar que CloudFormation criará recursos do IAM.
   b. Marque a caixa de seleção para confirmar que CloudFormation criará recursos do IAM com nomes personalizados.

c. Marque a caixa de seleção `CAPABILITY_AUTO_EXPAND` para confirmar a nova pilha. Para obter mais informações, consulte CreateStack na Referência de APIs do AWS CloudFormation .

5. Selecione Criar pilha.
6. Monitore o status da sua pilha. Você pode se conectar ao cluster depois que o status da pilha for`CREATE_COMPLETE`.

Conecte-se a um cluster AWS PCS criado com CloudFormation

Depois de criar um cluster AWS PCS a partir de um CloudFormation modelo, você pode usar o console AWS PCS (no Console de gerenciamento da AWS) para administrar o cluster. Você também pode se conectar a um dos nós de login do cluster para administrar o cluster, executar trabalhos e gerenciar dados. A CloudFormation pilha fornece links que você pode usar para se conectar ao seu cluster.

Para se conectar ao seu cluster

1. Abra o console do CloudFormation.
2. Escolha a pilha que você criou.
3. Escolha a guia Saídas da pilha.

   A pilha fornece os seguintes links:

   - PcsConsoleUrl— Escolha este link para abrir o console AWS PCS com o cluster selecionado. Você pode usá-lo para explorar as configurações de cluster, grupo de nós e fila.
   - Ec2 ConsoleUrl — Escolha este link para abrir o console do Amazon EC2, filtrado para mostrar as instâncias que o grupo de nós de login do cluster gerencia.

     Nessa visualização, você pode selecionar uma instância e escolher Connect. A instância do cluster de amostra oferece suporte a SSH de entrada e AWS Systems Manager conexões em um navegador da web. Para obter mais informações, consulte Conecte-se ao seu cluster AWS PCS.

     Depois de se conectar a uma instância de login, você pode seguir o tutorial emExplore o ambiente de cluster no AWS PCS.

Limpe um cluster AWS PCS em CloudFormation

Se você CloudFormation costumava criar seu cluster AWS PCS, você pode abrir o CloudFormation console e excluir a pilha para excluir o cluster e todos os recursos associados.

> **Important**
>
> Para o cluster de amostra, se você criou grupos ou filas de nós de computação adicionais em seu cluster (além dos `compute-1` grupos `login` e criados pelo CloudFormation modelo de amostra), você deve usar o console AWS PCS ou AWS CLI excluir esses recursos antes de excluir a CloudFormation pilha. Para obter mais informações, consulte Excluindo um cluster no AWS PCS.

Partes de um CloudFormation modelo para AWS PCS

Um CloudFormation modelo tem 1 ou mais seções, cada uma com uma finalidade específica. CloudFormation define formato, sintaxe e linguagem padrão em um modelo. Para obter mais informações, consulte Como trabalhar com CloudFormation modelos no Guia AWS CloudFormation do usuário.

CloudFormation os modelos são altamente personalizáveis e, portanto, seus formatos podem variar. Para entender as partes necessárias de um CloudFormation modelo para criar um cluster AWS PCS, recomendamos que você examine o modelo de amostra que fornecemos para criar um cluster de amostra. Este tópico explica resumidamente as seções desse modelo de amostra.

> **Important**
>
> Os exemplos de código neste tópico não estão completos. A presença de ellipsis (`[...]`) indica que há um código adicional que não é exibido. Para baixar o CloudFormation modelo completo em formato YAML, consulte. CloudFormation modelos para criar um cluster AWS PCS de amostra

Sumário

- Parâmetros
- Mapeamentos
- Recursos
- Saídas

Cabeçalho

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: AWS Parallel Computing Service "getting started" cluster
```

`AWSTemplateFormatVersion`identifica a versão do formato do modelo com a qual o modelo está em conformidade. Para obter mais informações, consulte a sintaxe da versão do formato de CloudFormation modelo no Guia do AWS CloudFormation usuário.

`Transform`especifica uma macro que é CloudFormation usada para processar o modelo. Para obter mais informações, consulte a seção Transformação do CloudFormation modelo no Guia AWS CloudFormation do usuário. A `AWS::Serverless-2016-10-31` transformação permite CloudFormation processar um modelo escrito na sintaxe AWS Serverless Application Model (AWS SAM). Para obter mais informações, consulte `AWS::Serverless`transform no Guia AWS CloudFormation do usuário.

Metadados

```
### Stack metadata
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: PCS Cluster configuration
        Parameters:
          - SlurmVersion
          - ManagedAccounting
          - AccountingPolicyEnforcement
      - Label:
          default: PCS ComputeNodeGroups configuration
        Parameters:
          - NodeArchitecture
          - KeyName
```

```
          - ClientIpCidr
      - Label:
          default: HPC Recipes configuration
        Parameters:
          - HpcRecipesS3Bucket
          - HpcRecipesBranch
```

A `metadata` seção de um CloudFormation modelo fornece informações sobre o próprio modelo. O modelo de amostra cria um cluster completo de computação de alto desempenho (HPC) que usa AWS PCS. A seção de metadados do modelo de amostra declara parâmetros que controlam como CloudFormation inicia (provisiona) a pilha correspondente. Existem parâmetros que controlam a escolha da arquitetura (`NodeArchitecture`), a versão do Slurm (`SlurmVersion`) e os controles de acesso (`KeyNameeClientIpCidr`).

Parâmetros

A `Parameters` seção define os parâmetros personalizados para o modelo. CloudFormation usa essas definições de parâmetros para criar e validar o formulário com o qual você interage ao iniciar uma pilha a partir desse modelo.

```
Parameters:

  NodeArchitecture:
    Type: String
    Default: x86
    AllowedValues:
      - x86
      - Graviton
    Description: Processor architecture for the login and compute node instances

  SlurmVersion:
    Type: String
    Default: 25.05
    Description: Version of Slurm to use
    AllowedValues:
         - 24.11
         - 25.05

  ManagedAccounting:
    Type: String
    Default: 'disabled'
    AllowedValues:
```

```
      - 'enabled'
      - 'disabled'
    Description: Monitor cluster usage, manage access control, and enforce resource
limits with Slurm accounting. Requires Slurm 24.11 or newer.

  AccountingPolicyEnforcement:
    Description: Specify which Slurm accounting policies to enforce
    Type: String
    Default: none
    AllowedValues:
      - none
      - 'associations,limits,safe'

  KeyName:
    Description: SSH keypair to log in to the head node
    Type: AWS::EC2::KeyPair::KeyName
    AllowedPattern: ".+"  # Required

  ClientIpCidr:
    Description: IP(s) allowed to access the login node over SSH. We recommend that
you restrict it with your own IP/subnet (x.x.x.x/32 for your own ip or x.x.x.x/24 for
range. Replace x.x.x.x with your own PUBLIC IP. You can get your public IP using tools
such as https://ifconfig.co/)
    Default: 127.0.0.1/32
    Type: String
    AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
    ConstraintDescription: Value must be a valid IP or network range of the form
x.x.x.x/x.

  HpcRecipesS3Bucket:
    Type: String
    Default: aws-hpc-recipes
    Description: HPC Recipes for AWS S3 bucket
    AllowedValues:
         - aws-hpc-recipes
         - aws-hpc-recipes-dev
  HpcRecipesBranch:
    Type: String
    Default: main
    Description: HPC Recipes for AWS release branch
    AllowedPattern: '^(?!.*/\.git$)(?!.*/\.)(?!.*\\.\.)[a-zA-Z0-9-_\.]+$'
```

Mapeamentos

A `Mappings` seção define pares de valores-chave que especificam valores com base em determinadas condições ou dependências.

```
Mappings:

  Architecture:
    AmiArchParameter:
      Graviton: arm64
      x86: x86_64
    LoginNodeInstances:
      Graviton: c7g.xlarge
      x86: c6i.xlarge
    ComputeNodeInstances:
      Graviton: c7g.xlarge
      x86: c6i.xlarge
```

Recursos

A `Resources` seção declara os AWS recursos a serem provisionados e configurados como parte da pilha.

```
Resources:

  [...]
```

O modelo provisiona a infraestrutura de cluster de amostra em camadas. Tudo começa com a `Networking` configuração da VPC. O armazenamento é fornecido por sistemas duplos: `EfsStorage` para armazenamento compartilhado e `FSxLStorage` para armazenamento de alto desempenho. O cluster principal é estabelecido por meio de`PCSCluster`.

```

  Networking:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        ProvisionSubnetsC: "False"
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/net/hpc_large_scale/assets/main.yaml'
```

```
  EfsStorage:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        SubnetIds: !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        SubnetCount: 1
        VpcId: !GetAtt [ Networking, Outputs.VPC ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/storage/efs_simple/assets/main.yaml'

  FSxLStorage:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        PerUnitStorageThroughput: 125
        SubnetId: !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        VpcId: !GetAtt [ Networking, Outputs.VPC ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/storage/fsx_lustre/assets/persistent.yaml'

  [...]

  # Cluster
  PCSCluster:
    Type: AWS::PCS::Cluster
    Properties:
      Name: !Sub '${AWS::StackName}'
      Size: SMALL
      Scheduler:
        Type: SLURM
        Version: !Ref SlurmVersion
      Networking:
        SubnetIds:
          - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        SecurityGroupIds:
          - !GetAtt [ PCSSecurityGroup, Outputs.ClusterSecurityGroupId ]
```

Para recursos de computação, o modelo cria dois grupos de nós: `PCSNodeGroupLogin` para um único nó de login e `PCSNodeGroupCompute` para até quatro nós de computação. Esses grupos de nós são suportados `PCSInstanceProfile` por permissões e, `PCSLaunchTemplate` por exemplo, configurações.

```
  # Compute Node groups
  PCSInstanceProfile:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        # We have to regionalize this in case CX use the template in more than one
 region. Otherwise,
        # the create action will fail since instance-role-${AWS::StackName} already
 exists!
        RoleName: !Sub '${AWS::StackName}-${AWS::Region}'
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/pcs-iip-minimal.yaml'

  PCSLaunchTemplate:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        VpcDefaultSecurityGroupId: !GetAtt [ Networking, Outputs.SecurityGroup ]
        ClusterSecurityGroupId: !GetAtt [ PCSSecurityGroup,
 Outputs.ClusterSecurityGroupId ]
        SshSecurityGroupId: !GetAtt [ PCSSecurityGroup,
 Outputs.InboundSshSecurityGroupId ]
        EfsFilesystemSecurityGroupId: !GetAtt [ EfsStorage, Outputs.SecurityGroupId ]
        FSxLustreFilesystemSecurityGroupId: !GetAtt [ FSxLStorage,
 Outputs.FSxLustreSecurityGroupId ]
        SshKeyName: !Ref KeyName
        EfsFilesystemId: !GetAtt [ EfsStorage, Outputs.EFSFilesystemId ]
        FSxLustreFilesystemId: !GetAtt [ FSxLStorage, Outputs.FSxLustreFilesystemId ]
        FSxLustreFilesystemMountName: !GetAtt [ FSxLStorage,
 Outputs.FSxLustreMountName ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/cfn-pcs-lt-efs-fsxl.yaml'

  # Compute Node groups - Login Nodes
  PCSNodeGroupLogin:
    Type: AWS::PCS::ComputeNodeGroup
    Properties:
      ClusterId: !GetAtt [PCSCluster, Id]
      Name: login
      ScalingConfiguration:
        MinInstanceCount: 1
        MaxInstanceCount: 1
```

```
      IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
      CustomLaunchTemplate:
        TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.LoginLaunchTemplateId ]
        Version: 1
      SubnetIds:
        - !GetAtt [ Networking, Outputs.DefaultPublicSubnet ]
      AmiId: !GetAtt [PcsSampleAmi, AmiId]
      InstanceConfigs:
        - InstanceType: !FindInMap [ Architecture, LoginNodeInstances, !Ref
NodeArchitecture ]

 # Compute Node groups - Compute Nodes
 PCSNodeGroupCompute:
   Type: AWS::PCS::ComputeNodeGroup
   Properties:
     ClusterId: !GetAtt [PCSCluster, Id]
     Name: compute-1
     ScalingConfiguration:
       MinInstanceCount: 0
       MaxInstanceCount: 4
     IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
     CustomLaunchTemplate:
       TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.ComputeLaunchTemplateId ]
       Version: 1
     SubnetIds:
       - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
     AmiId: !GetAtt [PcsSampleAmi, AmiId]
     InstanceConfigs:
       - InstanceType: !FindInMap [ Architecture, ComputeNodeInstances, !Ref
NodeArchitecture ]
```

O agendamento de trabalhos é feito por completo. PCSQueueCompute

```
 PCSQueueCompute:
   Type: AWS::PCS::Queue
   Properties:
     ClusterId: !GetAtt [PCSCluster, Id]
     Name: demo
     ComputeNodeGroupConfigurations:
       - ComputeNodeGroupId: !GetAtt [PCSNodeGroupCompute, Id]
```

A seleção da AMI acontece automaticamente por meio da função Pcs AMILookup Fn Lambda e dos recursos relacionados.

```
  PcsAMILookupRole:
    Type: AWS::IAM::Role
    [...]

  PcsAMILookupFn:
    Type: AWS::Lambda::Function
    Properties:
      Runtime: python3.12
      Handler: index.handler
      Role: !GetAtt PcsAMILookupRole.Arn
      Code:
        [...]
      Timeout: 30
      MemorySize: 128

  # Example of using the custom resource to look up an AMI
  PcsSampleAmi:
    Type: Custom::AMILookup
    Properties:
      ServiceToken: !GetAtt PcsAMILookupFn.Arn
      OperatingSystem: 'amzn2'
      Architecture: !FindInMap [ Architecture, AmiArchParameter, !Ref
 NodeArchitecture ]
      SlurmVersion: !Ref SlurmVersion
```

Saídas

O modelo gera a identificação e o gerenciamento URLs do cluster por meio de `ClusterIdPcsConsoleUrl`, e. `Ec2ConsoleUrl`

```
Outputs:
  ClusterId:
    Description: The Id of the PCS cluster
    Value: !GetAtt [ PCSCluster, Id ]

  PcsConsoleUrl:
    Description: URL to access the cluster in the PCS console
    Value: !Sub
```

```
      - https://${ConsoleDomain}/pcs/home?region=${AWS::Region}#/clusters/${ClusterId}
      - { ConsoleDomain: !If [ GovCloud, 'console.amazonaws-us-gov.com', !If [ China,
 'console.amazonaws.cn', !Sub '${AWS::Region}.console.aws.amazon.com']],
          ClusterId: !GetAtt [ PCSCluster, Id ]
        }
    Export:
      Name: !Sub ${AWS::StackName}-PcsConsoleUrl

  Ec2ConsoleUrl:
    Description: URL to access instance(s) in the login node group via Session Manager
    Value: !Sub
      - https://${ConsoleDomain}/ec2/home?region=
${AWS::Region}#Instances:instanceState=running;tag:aws:pcs:compute-node-group-id=
${NodeGroupLoginId}
      - { ConsoleDomain: !If [ GovCloud, 'console.amazonaws-us-gov.com', !If [ China,
 'console.amazonaws.cn', !Sub '${AWS::Region}.console.aws.amazon.com']],
          NodeGroupLoginId: !GetAtt [ PCSNodeGroupLogin, Id ]
        }
    Export:
      Name: !Sub ${AWS::StackName}-Ec2ConsoleUrl
```

CloudFormation modelos para criar um cluster AWS PCS de amostra

Região da AWS nome	Região da AWS	Exibir fonte	Pilha de lançamento
Leste dos EUA (Norte da Virgínia)	us-east-1	Baixar YAML	
Leste dos EUA (Ohio)	us-east-2	Baixar YAML	
Oeste dos EUA (Oregon)	us-west-2	Baixar YAML	
Ásia-Pacífico (Singapura)	ap-southeast-1	Baixar YAML	
Ásia-Pacífico (Sydney)	ap-southeast-2	Baixar YAML	

Região da AWS nome	Região da AWS	Exibir fonte	Pilha de lançamento
Ásia-Pacífico (Tóquio)	ap-northeast-1	Baixar YAML	
Europa (Frankfurt)	eu-central-1	Baixar YAML	
Europa (Irlanda)	eu-west-1	Baixar YAML	
Europa (Londres)	eu-west-2	Baixar YAML	
Europa (Estocolmo)	eu-north-1	Baixar YAML	
AWS GovCloud (Leste dos EUA)	us-gov-east-1	Baixar YAML	
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	Baixar YAML	

AWS Clusters PCS

Um cluster AWS PCS consiste nos seguintes componentes:

- Instâncias gerenciadas do software programador do sistema HPC, como o daemon de controle Slurm (). `slurmctld`
- Componentes que se integram ao programador do sistema HPC para provisionar e gerenciar instâncias da Amazon EC2.
- Componentes que se integram ao programador do sistema HPC para transmitir registros e métricas para a Amazon. CloudWatch

Esses componentes são executados em uma conta gerenciada por AWS. Eles trabalham juntos para gerenciar as EC2 instâncias da Amazon em sua conta de cliente. AWS O PCS provisiona interfaces de rede elásticas em sua sub-rede Amazon VPC para fornecer conectividade do software agendador às EC2 instâncias da Amazon (por exemplo, para oferecer suporte ao agendamento de trabalhos em lote nelas e permitir que os usuários executem comandos do agendador para listar e gerenciar esses trabalhos).

Tópicos

Criando um cluster no AWS PCS

Este tópico fornece uma visão geral das opções disponíveis e descreve o que considerar ao criar um cluster no Serviço de Computação AWS Paralela (AWS PCS). Se esta é a primeira vez que você cria um cluster AWS PCS, recomendamos que você sigaComece a usar o serviço de computação AWS paralela. O tutorial pode ajudá-lo a criar um sistema HPC funcional sem expandir para todas as opções disponíveis e arquiteturas de sistema possíveis.

> **Note**
>
> Depois de criar um cluster, você pode modificar várias configurações sem reconstruir sua infraestrutura. Para obter mais informações, consulte Atualizando um cluster no AWS PCS.

> **Note**
>
> Você pode definir configurações personalizadas do Slurm para implementar políticas avançadas de agendamento e gerenciamento de recursos. Para obter mais informações, consulte Definindo configurações personalizadas do Slurm no PCS AWS.

Pré-requisitos

- Uma VPC e uma sub-rede existentes que atendem aos requisitos. AWS Rede PCS Antes de implantar um cluster para uso em ambientes de produção, convém ter uma compreensão integral dos requisitos da VPC e da sub-rede. Para criar uma VPC e uma sub-rede, consulte. Criação de uma VPC para seu AWS cluster PCS
- Um diretor do IAM com permissões para criar e gerenciar recursos do AWS PCS. Para obter mais informações, consulte Identity and Access Management for AWS Parallel Computing Service.

Crie um cluster AWS PCS

Você pode usar o Console de gerenciamento da AWS ou AWS CLI para criar um cluster.

Console de gerenciamento da AWS

Para criar um cluster

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/home#/clusters e escolha Create cluster.
2. Na seção Configuração do cluster, insira os seguintes campos:
   - Nome do cluster — Um nome para seu cluster. O nome só pode conter caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um

caractere alfabético e não pode ter mais de 40 caracteres. O nome deve ser exclusivo no Região da AWS e no Conta da AWS qual você está criando o cluster.

- Agendador — Escolha um agendador e uma versão. Para obter mais informações, consulte Versões Slurm no PCS AWS.
- Tamanho do controle — Escolha um tamanho para o controle. Isso determina quantos trabalhos e nós de computação simultâneos podem ser gerenciados pelo cluster AWS PCS. Você só pode definir o tamanho do controlador quando o cluster é criado. Para obter mais informações sobre dimensionamento, consulteTamanho do cluster no AWS PCS.

3. Na seção Rede, selecione valores para os seguintes campos:

   - Tipo de rede — Escolha o tipo de endereço IP para seu cluster. Seu cluster pode usar um IPv4 ou IPv6, mas não os dois. A VPC e as sub-redes devem usar o mesmo tipo de endereço de rede. O bloco de endereços IP que você usa para cada sub-rede deve ter pelo menos 1 endereço disponível. AWS reserva alguns dos endereços em cada sub-rede. Para obter mais informações, consulte Blocos CIDR de sub-redes no Guia do usuário da Amazon VPC.
   - VPC — Escolha uma VPC existente que atenda aos requisitos da PCS. AWS Para obter mais informações, consulte AWS Requisitos e considerações sobre PCS, VPC e sub-rede. Depois de criar o cluster, você não pode alterar sua VPC. Se nenhum VPCs estiver listado, você deverá criar um primeiro.
   - Sub-rede — Todas as sub-redes disponíveis na VPC selecionada são listadas. Escolha uma sub-rede que atenda aos requisitos de sub-rede do AWS PCS. Para obter mais informações, consulte AWS Requisitos e considerações sobre PCS, VPC e sub-rede. Recomendamos que você selecione uma sub-rede privada para evitar a exposição dos endpoints do agendador à Internet pública.
   - Grupos de segurança — especifique os grupos de segurança que você deseja que o AWS PCS associe às interfaces de rede que ele cria para seu cluster. Você deve selecionar pelo menos um grupo de segurança que permita a comunicação entre seu cluster e seus nós de computação. Você pode selecionar Criar rapidamente um grupo de segurança para que o AWS PCS crie um com a configuração necessária na VPC selecionada ou selecione um grupo de segurança existente. Para obter mais informações, consulte Requisitos e considerações do grupo de segurança.

4. (Opcional) Na seção Configuração da contabilidade do Slurm, você pode ativar a contabilidade do Slurm e definir os parâmetros contábeis. Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.

5. (Opcional) Na seção Configuração do Slurm, você pode adicionar pares de nome e valor do parâmetro para definir configurações adicionais do Slurm. Para obter uma lista completa dos parâmetros compatíveis, consulteConfigurações personalizadas do Slurm para AWS clusters PCS.
6. (Opcional) Em Tags, adicione qualquer tag ao seu cluster AWS PCS.
7. Selecione Criar cluster. O campo Status é exibido `Creating` enquanto o AWS PCS cria o cluster. Esse processo pode levar alguns minutos.

> **Important**
>
> Só pode haver 1 cluster em um `Creating` estado Região da AWS por pessoa Conta da AWS. AWS O PCS retornará um erro se já houver um cluster em um `Creating` estado quando você tentar criar um cluster.

AWS CLI

Para criar um cluster

1. Crie o cluster usando o comando a seguir. Antes da execução do comando, realize as seguintes substituições:
   - *`region`*Substitua pelo ID do Região da AWS qual você deseja criar seu cluster, como`us-east-1`.
   - Substitua *`my-cluster`* por um nome de cluster. O nome só pode conter caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um caractere alfabético e não pode ter mais de 40 caracteres. O nome deve ser exclusivo dentro Região da AWS e Conta da AWS onde você está criando o cluster.
   - *`25.05`*Substitua por qualquer versão compatível do Slurm.

     > **Note**
     >
     > AWS Atualmente, o PCS suporta Slurm 25.05 e 24.11.

   - *`SMALL`*Substitua por qualquer tamanho de cluster compatível. Isso determina quantos trabalhos e nós de computação simultâneos podem ser gerenciados pelo cluster AWS

PCS. Ele só pode ser definido quando o cluster é criado. Para obter mais informações sobre dimensionamento, consulteTamanho do cluster no AWS PCS.

- Substitua o valor `subnetIds` por pelo seu. Recomendamos que você selecione uma sub-rede privada para evitar a exposição dos endpoints do agendador à Internet pública.
- Especifique o `securityGroupIds` que você deseja que o AWS PCS associe às interfaces de rede que ele cria para seu cluster. Os grupos de segurança devem estar na mesma VPC do cluster. Você deve selecionar pelo menos um grupo de segurança que permita a comunicação entre seu cluster e seus nós de computação. Para obter mais informações, consulte Requisitos e considerações do grupo de segurança.

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=25.05 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

- para usar IPv6, adicione `networkType=IPV6` à `--networking` configuração.

```
  --networking networkType=IPV6,subnetIds=subnet-
ExampleId1,securityGroupIds=sg-ExampleId1
```

- Opcionalmente, você pode adicionar a `--slurm-configration` opção de personalizar o comportamento do Slurm e especificar as opções de configuração do Slurm. O exemplo a seguir define o tempo de inatividade de redução para 60 minutos (3600 segundos), ativa a contabilização do Slurm e especifica `slurm.conf` as configurações como o valor de. `slurmCustomSettings` Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.

> **Note**
>
> A contabilidade é compatível com o Slurm 24.11 ou posterior.

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=25.05 \
```

```
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
    --slurm-configuration
scaleDownIdleTimeInSeconds=3600,accounting='{mode=STANDARD}',slurmCustomSettings='[{p
```

2. O provisionamento do cluster pode levar vários minutos. Você pode consultar o status do cluster com o comando a seguir. Não continue criando filas ou grupos de nós de computação até que o campo de status do cluster seja exibido. `ACTIVE`

```
aws pcs get-cluster --region region --cluster-identifier my-cluster
```

> **Important**
>
> Só pode haver 1 cluster em um `Creating` estado Região da AWS por pessoa Conta da AWS. AWS O PCS retornará um erro se já houver um cluster em um `Creating` estado quando você tentar criar um cluster.

Próximas etapas recomendadas para seu cluster

- Adicione grupos de nós de computação.
- Adicione filas.
- Ativar o registro em log.

Atualizando um cluster no AWS PCS

AWS O PCS permite que você atualize as configurações do cluster após a criação por meio da UpdateCluster API ou do console. Você pode modificar as configurações do cluster sem reconstruir sua infraestrutura, o que reduz a sobrecarga operacional e minimiza as interrupções.

Benefícios das atualizações de cluster

A atualização dos clusters AWS PCS permite que você adapte a infraestrutura de HPC aos novos requisitos sem interromper o serviço. As alterações na configuração levam minutos em vez da hora ou mais necessária para reconstruir clusters. Esse recurso é importante para ambientes de produção que exigem tempo mínimo de inatividade e para equipes que precisam ajustar as configurações do cluster à medida que os padrões de carga de trabalho mudam.

Alterações de configuração suportadas

Você pode modificar três categorias principais de configurações:

- Configuração contábil - ative ou desative a contabilidade gerenciada e defina as configurações de retenção.
- Comportamento de redução de escala - ajuste o `scaleDownIdleTime` parâmetro, que controla por quanto tempo as instâncias dinâmicas permanecem inativas antes que o AWS PCS as encerre automaticamente.
- Configurações personalizadas do Slurm - modifique qualquer uma das configurações do Slurm suportadas que se aplicam no nível do cluster, incluindo Prolog, Epilog e. SelectTypeParameters

Limitações

Você não pode modificar determinadas configurações após a criação do cluster. Isso inclui:

- Configurações do grupo de segurança
- Seleção de sub-rede VPC
- Tamanho do cluster
- Versão Slurm
- Nome do cluster

Essas configurações são fundamentais para a arquitetura do cluster e exigem a criação de um novo cluster para modificá-las.

Pré-requisitos para atualizações de cluster

Antes de atualizar um cluster, verifique se as seguintes condições foram atendidas:

- O cluster deve estar em `ACTIVEUPDATE_FAILED`, ou `SUSPENDED` estado
- Todos os recursos associados (filas, grupos de nós de computação) devem estar no estado `ACTIVE`
- Você deve ter as permissões apropriadas do IAM para a UpdateCluster operação
- Nenhuma outra operação de atualização pode estar em andamento

Processo de atualização e impacto no trabalho

Durante uma operação de atualização, os nós de computação continuam executando trabalhos existentes mesmo quando o controlador de cluster fica brevemente inacessível. No entanto, o sistema não pode aceitar novos envios de trabalhos ou tomar decisões de agendamento durante esse período.

Você pode monitorar as atualizações do cluster por meio das interfaces do console e da API. O cluster passará pelos seguintes estados durante uma atualização:

- `UPDATING`- Atualização em andamento
- `ACTIVE`- Atualização concluída com sucesso
- `UPDATE_FAILED`- A atualização encontrou um erro

Faturamento durante as atualizações

As cobranças horárias padrão do seu cluster AWS PCS continuam durante as operações de atualização. Quando você atualiza um cluster para desativar a contabilização, a cobrança pelo recurso de contabilidade é interrompida assim que o cluster entra no `UPDATING` estado. Ao ativar a contabilidade, o faturamento não começa até que o cluster conclua com êxito a atualização e retorne ao `ACTIVE` estado.

Tópicos

- Atualizar um cluster AWS PCS
- Perguntas frequentes sobre a atualização de clusters no AWS PCS
- Solução de problemas de atualizações do cluster AWS PCS

Atualizar um cluster AWS PCS

Use essas etapas para modificar as configurações do agendador, a configuração contábil e as configurações personalizadas do Slurm em seu cluster. Para obter mais informações, consulte Configurações personalizadas do Slurm para AWS clusters PCS.

Pré-requisitos

- O cluster deve estar em `ACTIVE``UPDATE_FAILED`, ou `SUSPENDED` estado

- Todos os recursos associados (filas, grupos de nós de computação) devem estar no estado `ACTIVE`
- Nenhuma outra operação de atualização pode estar em andamento

Procedimento

Console de gerenciamento da AWS

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. No painel de navegação, escolha Clusters.
3. Selecione o cluster a ser atualizado.
4. Escolha Editar.
5. Na página Editar cluster, modifique as configurações desejadas:
   - Em Configuração do Scheduler, atualize o tempo ocioso Scale-down para controlar por quanto tempo as instâncias dinâmicas permanecem inativas antes do encerramento automático.
   - Modifique as configurações dos parâmetros do tipo Prolog, Epilog e Select conforme necessário.
   - Ative, desative ou configure o tempo de retenção para a contabilidade gerenciada.
   - Em Configurações adicionais do agendador, adicione, edite ou remova as configurações personalizadas do Slurm. Para obter mais informações sobre os parâmetros suportados, consulteConfigurações personalizadas do Slurm para AWS clusters PCS.

   > **Note**
   >
   > Os campos que não podem ser editados são exibidos somente para leitura e mostram seus valores atuais.
6. Escolha Atualizar para enviar as alterações.
7. Monitore o status do cluster, que aparece como “Atualização” durante o processo. O status muda quando a atualização é concluída com êxito.

AWS CLI

1. Abra um terminal ou prompt de comando.
2. Verifique o status do cluster usando o seguinte comando:

```
aws pcs get-cluster --cluster-identifier my-cluster
```

3. Envie uma solicitação de atualização usando um dos exemplos a seguir:

   - Para habilitar a contabilidade gerenciada:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration 'accounting={mode=STANDARD}'
```

   - Para atualizar uma configuração do Slurm Prolog:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/
prolog.sh"}]'
```

   - Para atualizar o tempo ocioso de redução:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration 'scaleDownIdleTimeInSeconds=300'
```

4. Monitore o progresso da atualização verificando o status do cluster:

```
aws pcs get-cluster --cluster-identifier my-cluster
```

Depois de uma solicitação de atualização bem-sucedida, o comando retorna o objeto Cluster com todas as alterações. O status do cluster muda de `UPDATING` para `ACTIVE` quando concluído.

Perguntas frequentes sobre a atualização de clusters no AWS PCS

Obtenha respostas para perguntas comuns sobre a atualização de configurações de cluster no AWS PCS.

Quais configurações posso modificar?

Você pode modificar a configuração contábil (ativar/desativar a contabilidade gerenciada), o comportamento de redução (parâmetro scaleDownIdle Time) e qualquer uma das configurações personalizadas do Slurm suportadas que se aplicam no nível do cluster. Você não pode modificar grupos de segurança, sub-redes VPC, tamanho do cluster, versão do Slurm ou nome do cluster.

Posso colocar várias atualizações na fila?

Não. Você deve esperar que o cluster retorne ao `ACTIVE` estado antes de enviar outra atualização. Todos os recursos associados (filas, grupos de nós de computação) também devem estar no `ACTIVE` estado.

Posso cancelar uma operação de atualização do cluster?

Não, você não pode cancelar uma operação de atualização de cluster em andamento.

Posso enviar trabalhos enquanto meu cluster está sendo atualizado?

Recomendamos que você evite enviar trabalhos durante as atualizações do cluster. O controlador Slurm pode estar indisponível durante o processo de atualização.

Meus trabalhos continuarão sendo executados durante as atualizações do cluster?

Sim, os trabalhos em execução continuam sendo executados nos nós de computação mesmo quando o controlador de cluster fica brevemente inacessível durante o processo de atualização. No entanto, o status do trabalho pode não ser atualizado até que o controlador fique disponível novamente.

Como o faturamento é afetado durante as atualizações?

As cobranças horárias padrão continuam durante as operações de atualização. Ao desativar a contabilidade, o faturamento é interrompido quando o cluster entra em `UPDATING` estado. Ao ativar a contabilidade, o faturamento começa quando o cluster retorna ao `ACTIVE` estado com sucesso.

Solução de problemas de atualizações do cluster AWS PCS

Este tópico ajuda você a identificar e resolver problemas comuns que podem ocorrer ao atualizar as configurações do cluster.

Falha na atualização com erro de configuração contábil

Causa comum

O cluster entra no `UPDATE_FAILED` estado e a mensagem de erro indica um problema de configuração contábil. Isso geralmente ocorre quando a configuração contábil é incompatível com a versão atual do Slurm ou contém configurações inválidas.

Resolução

Revise suas configurações de contabilidade para verificar a compatibilidade com a versão do Slurm do seu cluster e envie uma solicitação de atualização corrigida com parâmetros de configuração válidos.

Falha na atualização com erro de configurações personalizadas

Causa comum

O cluster entra no `UPDATE_FAILED` estado e a mensagem de erro indica um problema nas configurações personalizadas do Slurm. Isso ocorre quando você fornece valores de parâmetros inválidos do Slurm ou combinações de parâmetros não suportadas.

Resolução

Valide suas configurações personalizadas do Slurm em relação aos parâmetros compatíveis e envie uma solicitação de atualização corrigida com valores e combinações de parâmetros válidos.

Não é possível enviar a solicitação de atualização

Causa comum

O botão de atualização está desativado no console ou a API retorna um erro de nível 400. Isso ocorre quando o cluster não está em um estado apropriado, os recursos associados não estão ativos ou há falhas de validação em sua configuração.

Resolução

Aguarde até que o cluster e todos os recursos associados atinjam o `ACTIVE` estado e, em seguida, revise sua configuração em busca de erros de validação antes de reenviar a solicitação de atualização.

Erros de validação

Causa comum

O comando retorna imediatamente com um erro HTTP de 400 níveis e uma mensagem descritiva. Isso ocorre devido ao estado do cluster, ao estado do recurso ou aos parâmetros de configuração inválidos.

Resolução

Solucione o erro de validação específico mencionado na resposta e repita a operação de atualização.

Excluindo um cluster no AWS PCS

Este tópico fornece uma visão geral de como excluir um cluster do AWS PCS.

Considerações ao excluir um AWS cluster PCS

- Todas as filas associadas ao cluster devem ser excluídas antes que o cluster possa ser excluído. Para obter mais informações, consulte Excluindo uma fila no PCS AWS.
- Todos os grupos de nós de computação associados ao cluster devem ser excluídos antes que o cluster possa ser excluído. Para obter mais informações, consulte Excluindo um grupo de nós de computação no PCS AWS.

Excluir o cluster

Você pode usar o Console de gerenciamento da AWS ou AWS CLI para excluir um cluster.

Console de gerenciamento da AWS

Para excluir um cluster

1. Abra o console AWS PCS.
2. Selecione o cluster a ser excluído.
3. Escolha Excluir.
4. O campo Status do cluster é exibido`Deleting`. Pode demorar vários minutos para isso ser concluído.

AWS CLI

Para excluir um cluster

1. Use o comando a seguir para excluir um cluster, com essas substituições:
   - *region-code*Substitua por aquele em que Região da AWS seu cluster está.
   - *my-cluster*Substitua pelo nome ou ID do seu cluster.

```
aws pcs delete-cluster --region region-code --cluster-identifier my-cluster
```

2. A exclusão do cluster pode levar alguns minutos. Você pode verificar o status do seu cluster com o comando a seguir.

```
aws pcs get-cluster --region region-code --cluster-identifier my-cluster
```

Tamanho do cluster no AWS PCS

AWS O PCS fornece clusters altamente disponíveis e seguros, ao mesmo tempo em que automatiza tarefas importantes, como aplicação de patches, provisionamento de nós e atualizações.

Ao criar um cluster, você seleciona um tamanho para ele com base em dois fatores:

- O número de nós de computação que ele gerenciará
- O número de trabalhos ativos e em fila que você espera executar no cluster

> **Important**
>
> Você não pode alterar o tamanho do cluster depois de criar o cluster. Se você precisar alterar o tamanho, deverá criar um novo cluster.

Tamanho do cluster do Slurm	Número de instâncias gerenciadas	Número de trabalhos ativos e em fila
Small	Até 32	Até 256

Tamanho do cluster do Slurm	Número de instâncias gerenciadas	Número de trabalhos ativos e em fila
Médio	Até 512	Até 8192
Grande	Até 2048	Até 16384

Exemplos

- Se seu cluster tiver até 24 instâncias gerenciadas e executar até 100 trabalhos, escolha Pequeno.
- Se seu cluster tiver até 24 instâncias gerenciadas e executar até 1.000 trabalhos, escolha Médio.
- Se seu cluster tiver até 1.000 instâncias gerenciadas e executar até 100 trabalhos, escolha Grande.
- Se seu cluster tiver até 1.000 instâncias gerenciadas e executar até 10.000 trabalhos, escolha Grande.

Trabalhando com segredos de cluster no AWS PCS

Como parte da criação de um cluster, o AWS PCS cria um segredo de cluster que é necessário para se conectar ao agendador de tarefas no cluster. Você também cria grupos de nós de computação AWS PCS, que definem conjuntos de instâncias a serem executadas em resposta a eventos de escalabilidade. AWS O PCS configura instâncias iniciadas por esses grupos de nós de computação com o segredo do cluster para que eles possam se conectar ao agendador de tarefas. Há casos em que talvez você queira configurar os clientes do Slurm manualmente. Os exemplos incluem criar um nó de login persistente ou configurar um gerenciador de fluxo de trabalho com recursos de gerenciamento de tarefas.

AWS O PCS armazena o segredo do cluster como um segredo gerenciado com o prefixo `pcs!` in AWS Secrets Manager. O custo do segredo está incluído na cobrança pelo uso do AWS PCS. Você pode alternar os segredos do cluster AWS Secrets Manager para manter a conformidade de segurança e corrigir possíveis comprometimentos de segurança.

Tópicos

- Use AWS Secrets Manager para encontrar o segredo do cluster
- Use o AWS PCS para encontrar o segredo do cluster
- Obtenha o segredo do cluster Slurm

- Segredos do cluster rotativo no AWS PCS

Use AWS Secrets Manager para encontrar o segredo do cluster

Console de gerenciamento da AWS

1. Navegue até o console do Secrets Manager.
2. Escolha Segredos e pesquise o `pcs!` prefixo.

> **Note**
>
> Um segredo de cluster AWS PCS tem um nome no formato em `pcs!slurm-secret-`*`cluster-id`* que *`cluster-id`* é o ID do cluster AWS PCS.

AWS CLI

Cada segredo do cluster AWS PCS também é marcado com`aws:pcs:`*`cluster-id`*. Você pode obter o ID secreto de um cluster com o comando a seguir. Faça essas substituições antes de executar o comando:

- *`region`*Substitua pelo Região da AWS para criar seu cluster, com`us-east-1`.
- *`cluster-id`*Substitua pelo ID do cluster AWS PCS para encontrar o segredo do cluster.

```
aws secretsmanager list-secrets \
    --region region  \
    --filters Key=tag-key,Values=aws:pcs:cluster-id \
            Key=tag-value,Values=cluster-id
```

Use o AWS PCS para encontrar o segredo do cluster

Você pode usar o AWS CLI para encontrar o ARN de um segredo de cluster AWS PCS. Digite o comando a seguir, fazendo as seguintes substituições:

- *`region`*Substitua pelo Região da AWS para criar seu cluster, com`us-east-1`.
- *`my-cluster`*Substitua pelo nome ou identificador do seu cluster.

```
aws pcs get-cluster --region region --cluster-identifier my-cluster
```

O exemplo de saída a seguir é do `get-cluster` comando. Vocês podem usar `secretArn` e `secretVersion` juntos para descobrir o segredo.

```
{
    "cluster": {
        "name": "get-started",
        "id": "pcs_123456abcd",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
        "status": "ACTIVE",
        "createdAt": "2024-12-17T21:03:52+00:00",
        "modifiedAt": "2024-12-17T21:03:52+00:00",
        "scheduler": {
            "type": "SLURM",
            "version": "25.05"
        },
        "size": "SMALL",
        "slurmConfiguration": {
            "authKey": {
                "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!
slurm-secret-pcs_123456abcd-a12ABC",
                "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
            }
        },
        "networking": {
            "subnetIds": [
                "subnet-0123456789abcdef0"
            ],
            "securityGroupIds": [
                "sg-0123456789abcdef0"
            ]
        },
        "endpoints": [
            {
                "type": "SLURMCTLD",
                "privateIpAddress": "10.3.149.220",
                "port": "6817"
            }
        ]
    }
}
```

Obtenha o segredo do cluster Slurm

Você pode usar o Secrets Manager para obter a versão atual codificada em base64 de um segredo de cluster do Slurm. O exemplo a seguir usa o. AWS CLI Faça as seguintes substituições antes de executar o comando.

- *region*Substitua pelo Região da AWS para criar seu cluster, como`us-east-1`.
- *secret-arn*Substitua pelo `secretArn` de um cluster AWS PCS.

```
aws secretsmanager get-secret-value \
    --region region \
    --secret-id 'secret-arn' \
    --version-stage AWSCURRENT \
    --query 'SecretString' \
    --output text
```

Para obter informações sobre como usar o segredo do cluster Slurm, consulte. Usando instâncias autônomas como nós de login do AWS PCS

Permissões

Você usa um diretor do IAM para obter o segredo do cluster Slurm. O diretor do IAM deve ter permissão para ler o segredo. Para obter mais informações, consulte Termos e conceitos de funções no Guia AWS Identity and Access Management do usuário.

O exemplo de política do IAM a seguir permite o acesso a um exemplo de segredo de cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSecretValueRetrievalAndVersionListing",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:012345678901:secret:pcs!
slurm-secret-s3431v9rx2-FN7tJF"
        }
    ]
```

```
}
```

Segredos do cluster rotativo no AWS PCS

Use o AWS Secrets Manager Managed Rotation para alternar os segredos do cluster no AWS PCS. A rotação regular de segredos é uma prática recomendada de segurança para manter uma postura de segurança forte em ambientes de HPC. Esse recurso permite que você atenda aos padrões de conformidade do setor, incluindo HIPAA e FedRAMP, que exigem a rotação regular de credenciais.

O segredo do cluster tem dois propósitos: autenticar os nós de computação que se juntam ao cluster e ser a chave JWT para a autenticação da API REST do Slurm. Quando girados, os dois aspectos são afetados simultaneamente.

Como funciona a rotação secreta do cluster

Prepare-se manualmente para manter a estabilidade do cluster durante a rotação secreta:

1. Preparação — escale todos os grupos de nós de computação para 0 de capacidade e garanta que nenhuma tarefa esteja em execução
2. Rotação — Inicie a rotação por meio do console ou API do Secrets Manager
3. Monitoramento — Acompanhe o progresso por meio de CloudTrail eventos
4. Recuperação — redimensione os grupos de nós de computação até a capacidade desejada

Durante a rotação, seu cluster permanece no `ACTIVE` estado e o faturamento continua normalmente. O processo normalmente leva alguns minutos.

Requisitos e limitações

Antes de alternar os segredos do cluster, preencha estes requisitos:

- O cluster deve estar em `ACTIVE` ou `UPDATE_FAILED` estado
- A função do IAM deve ter `secretsmanager:RotateSecret` permissão
- Todos os grupos de nós de computação devem ser dimensionados para 0 de capacidade
- Pare todos os trabalhos antes da rotação

Limitações:

- Preparação manual necessária para cada rotação

- Os tokens JWT existentes se tornam inválidos e exigem reemissão
- Os nós de login BYO exigem atualização secreta manual após a rotação

Tópicos

Faça a rotação de um segredo de cluster no AWS PCS

Alterne o segredo do seu cluster para cumprir os requisitos de segurança e resolver possíveis comprometimentos. Esse processo exige colocar seu cluster em modo de manutenção.

Pré-requisitos

- Função do IAM com `secretsmanager:RotateSecret` permissão
- Cluster em `ACTIVE` ou `UPDATE_FAILED` estado

Procedimento

1. Notifique os usuários do cluster sobre a próxima janela de manutenção.
2. Coloque o cluster em modo de manutenção escalando todos os grupos de nós de computação para 0 de capacidade.
    a. Use a UpdateComputeNodeGroup API para definir ambos minInstanceCount e como 0 maxInstanceCount para todos os grupos de nós de computação.
    b. Espere até que todos os nós parem.
    c. Opcional: elimine as filas do agendador com os comandos do Slurm antes de encerrar a capacidade para um gerenciamento adequado do trabalho.
3. Inicie a rotação por meio do Secrets Manager.
    - Método de console:
        - Navegue até Secrets Manager, selecione o segredo do cluster e escolha Rotate secret.
    - Método de API:
        - Use a `rotate-secret` API Secrets Manager.

4. Monitore o progresso da rotação.

    a. Acompanhe o progresso por meio de CloudTrail eventos.

    b. Verifique `lastRotatedDate` no console do Secrets Manager ou na `secretsmanager:describeSecret` API.

    c. Aguarde `RotationSucceeded` nosso `RotationFailed` CloudTrail evento.

5. Após a rotação bem-sucedida, restaure a capacidade do cluster.

    a. Use a UpdateComputeNodeGroup API para redefinir os grupos de nós para a min/max capacidade desejada.

    b. Para nós de login AWS gerenciados por PCs: nenhuma ação adicional é necessária.

    c. Para nós de login BYO:

        i. Conecte-se aos nós de login.

        ii. Atualize `/etc/slurm/slurm.key` com o novo segredo do Secrets Manager.

        iii. Reinicie o Slurm Auth e o Cred Kiosk Daemon (sackd).

Perguntas frequentes sobre rotação secreta de cluster no AWS PCS

Encontre respostas para perguntas comuns sobre rotação secreta de cluster no AWS PCS.

O que é um segredo de cluster?

Um segredo de cluster é uma credencial segura que permite comunicações seguras entre o controlador Slurm e os nós de computação AWS PCS. Ela também serve como chave JSON Web Token (JWT) para autenticação da API Slurm REST.

Qual é a diferença entre o segredo do cluster e a chave JWT?

No AWS PCS, o segredo do cluster e a chave JWT são o mesmo recurso com propósitos diferentes. O segredo do cluster autentica as comunicações internas do Slurm, enquanto a chave JWT assina tokens para autenticação da API REST. Quando girados, os dois aspectos são afetados simultaneamente.

Quanto tempo demora a rotação?

O processo de rotação normalmente leva alguns minutos. Seu cluster permanece no estado ATIVO e o faturamento continua normalmente durante a rotação.

Posso programar rotações automáticas?

Você pode ativar a rotação programada no Secrets Manager. No entanto, a versão inicial requer preparação manual (escalando grupos de nós para 0) antes de cada rotação.

Meus tokens JWT existentes ainda funcionarão após a rotação?

Não, os tokens JWT existentes se tornam inválidos após a rotação. Emita novos tokens para clientes da API REST.

Onde posso encontrar o segredo do meu cluster?

Você pode encontrar o segredo do seu cluster no console Secrets Manager ou no console AWS PCS. Para obter instruções detalhadas, consulte Use AWS Secrets Manager para encontrar o segredo do cluster Use o AWS PCS para encontrar o segredo do cluster e.

Por que a rotação exige a escalabilidade dos grupos de nós para 0?

A rotação não requer instâncias em execução para garantir a estabilidade do cluster durante o processo de atualização secreta. Isso evita conflitos de autenticação entre segredos antigos e novos.

A quais requisitos de conformidade esse recurso oferece suporte?

Esse recurso permite que a AWS PCS atenda aos padrões de conformidade do setor, incluindo HIPAA e FedRAMP, que exigem a rotação regular de credenciais como parte de seus controles de segurança.

Solução de problemas de rotação secreta de cluster no AWS PCS

A rotação secreta do cluster falha se o ambiente não estiver preparado adequadamente. A causa mais comum são as instâncias ativas em seu cluster. Para evitar falhas:

1. Defina todos os grupos de nós para 0 de capacidade.
2. Aguarde até que os nós parem.
3. Verifique se seu cluster não está nos seguintes estados: `CREATE_FAILED`DELETE_FAILED,`RESUMING`,`SUSPENDING`, ou`SUSPENDED`.

Se a rotação falhar:

- Um RotationFailed CloudTrail evento aparece

- O segredo do cluster permanece inalterado
- Confira o RotationFailed evento CloudTrail para obter detalhes
- Conclua todas as etapas de preparação para uma rotação bem-sucedida

AWS Grupos de nós de computação PCS

Um grupo de nós de computação AWS PCS é uma coleção lógica de nós (EC2 instâncias da Amazon). Esses nós podem ser usados para executar trabalhos de computação, bem como para fornecer acesso interativo baseado em shell a um sistema de HPC. Um grupo de nós de computação consiste em regras para criar nós, incluindo quais tipos de EC2 instâncias da Amazon usar, quantas instâncias executar, se usar instâncias spot ou instâncias sob demanda, quais sub-redes e grupos de segurança usar e como configurar cada instância quando ela for iniciada. Quando essas regras são atualizadas, o AWS PCS atualiza os recursos associados ao grupo de nós de computação de acordo com a correspondência.

Tópicos

Criação de um grupo de nós de computação no AWS PCS

Este tópico fornece uma visão geral das opções disponíveis e descreve o que considerar ao criar um grupo de nós de computação no Serviço de Computação AWS Paralela (AWS PCS). Se esta é a primeira vez que você cria um grupo de nós de computação no AWS PCS, recomendamos que você siga o tutorial emComece a usar o serviço de computação AWS paralela. O tutorial pode ajudá-lo a criar um sistema HPC funcional sem expandir para todas as opções disponíveis e arquiteturas de sistema possíveis.

> **Note**
>
> Você pode definir configurações personalizadas do Slurm em grupos de nós de computação para controlar a utilização de recursos e os comportamentos em nível de nó. Para obter mais informações, consulte Definindo configurações personalizadas do Slurm no PCS AWS.

> **Important**
>
> AWS Atualmente, o PCS requer um kernel com IPv4 suporte para comunicação com nós locais, mesmo quando você usa o AWS PCS em uma rede IPv6 somente. Para obter mais informações, consulte Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

Pré-requisitos

- Cotas de serviço suficientes para iniciar o número desejado de instâncias do EC2 em seu. Região da AWS Você pode usar o Console de gerenciamento da AWSpara verificar e solicitar aumentos em suas cotas de serviço.
- Uma VPC e uma sub-rede existentes que atendem aos requisitos de rede do AWS PCS. Recomendamos que você entenda completamente esses requisitos antes de implantar um cluster para uso em produção. Para obter mais informações, consulte AWS Requisitos e considerações sobre PCS, VPC e sub-rede. Você também pode usar um CloudFormation modelo para criar uma VPC e sub-redes. AWS fornece uma receita de HPC para o CloudFormation modelo. Para obter mais informações, consulte aws-hpc-recipesem GitHub.
- Um perfil de instância do IAM com permissões para chamar a ação da `RegisterComputeNodeGroupInstance` API AWS PCS e acessar quaisquer outros AWS recursos necessários para suas instâncias de grupo de nós. Para obter mais informações, consulte Perfis de instância do IAM para o AWS Parallel Computing Service.
- Um modelo de lançamento para suas instâncias de grupos de nós. Para obter mais informações, consulte Usando modelos de lançamento do Amazon EC2 com PCS AWS.
- Para criar um grupo de nós computacionais que usa instâncias spot do Amazon EC2, você deve ter a função vinculada AWSServiceRoleForEC2ao serviço Spot em seu. Conta da AWS Para obter mais informações, consulte Função spot do Amazon EC2 para PCS AWS.

Crie um grupo de nós de computação no AWS PCS

Você pode criar um grupo de nós de computação usando o. Console de gerenciamento da AWS ou o. AWS CLI

Console de gerenciamento da AWS

Para criar seu grupo de nós de computação usando o console

1. Abra o console AWS PCS.
2. Selecione o cluster em que você deseja criar um grupo de nós de computação. Navegue até grupos de nós de computação e escolha Criar.
3. Na seção Configuração do grupo de nós de computação, forneça um nome para seu grupo de nós. O nome só pode conter caracteres alfanuméricos e hífens que diferenciem maiúsculas e minúsculas. Ele deve começar com um caractere alfabético e não pode ter mais de 25 caracteres. O nome deve ser exclusivo dentro do cluster.
4. Em Configuração de computação, insira ou selecione estes valores:

   a. Modelo de execução do EC2 — Selecione um modelo de execução personalizado para usar nesse grupo de nós. Os modelos de execução podem ser usados para personalizar configurações de rede, como sub-rede e grupos de segurança, configuração de monitoramento e armazenamento em nível de instância. Se você não tiver um modelo de lançamento preparado, consulte Usando modelos de lançamento do Amazon EC2 com PCS AWS para saber como criar um.

   > **Important**
   >
   > AWS O PCS cria um modelo de lançamento gerenciado para cada grupo de nós de computação. Esses são nomeados`pcs-`*`identifier`*`-do-not-delete`. Não os selecione ao criar ou atualizar um grupo de nós de computação, ou o grupo de nós não funcionará corretamente.

   b. Versão do modelo de lançamento do EC2 — Você deve selecionar uma versão do seu modelo de lançamento personalizado. Se você alterar a versão posteriormente, deverá atualizar o grupo de nós de computação para detectar alterações no modelo de execução. Para obter mais informações, consulte Atualização de um grupo de nós de computação AWS PCS.

   c. ID de AMI — se seu modelo de lançamento não incluir um ID de AMI ou se você quiser substituir o valor no modelo de lançamento, forneça um ID de AMI aqui. Observe que a AMI usada para o grupo de nós deve ser compatível com o AWS PCS. Você também pode selecionar uma amostra de AMI fornecida por AWS. Para obter mais informações sobre esse tópico, consulteAmazon Machine Images (AMIs) para AWS PCS.

d. Perfil de instância do IAM — escolha um perfil de instância para o grupo de nós. Um perfil de instância concede à instância permissões para acessar AWS recursos e serviços com segurança. Se você não tiver um preparado, você pode selecionar Criar um perfil básico para que o AWS PCS crie um para você com a política mínima, ou consulte Perfis de instância do IAM para o AWS Parallel Computing Service.

e. Sub-redes — Escolha uma ou mais sub-redes na VPC em que seu cluster PCS está implantado. AWS Se você selecionar várias sub-redes, as comunicações EFA não estarão disponíveis entre os nós, e a comunicação entre nós em sub-redes diferentes poderá aumentar a latência. Certifique-se de que as sub-redes especificadas aqui correspondam às que você define no modelo de execução do EC2.

f. Instâncias — escolha um ou mais tipos de instância para atender às solicitações de escalabilidade no grupo de nós. Todos os tipos de instância devem ter a mesma arquitetura de processador (x86_64 ou arm64) e número de v. CPUs Se as instâncias tiverem GPUs, todos os tipos de instância deverão ter o mesmo número de GPUs.

g. Configuração de escalabilidade — especifique o número mínimo e máximo de instâncias para o grupo de nós. Você pode definir uma configuração estática, na qual há um número fixo de nós em execução, ou uma configuração dinâmica, na qual até a contagem máxima de nós pode ser executada. Para uma configuração estática, defina o mínimo e o máximo para o mesmo número, maior que zero. Para uma configuração dinâmica, defina o mínimo de instâncias como zero e o máximo de instâncias como um número maior que zero. AWS O PCS não oferece suporte a grupos de nós de computação com uma combinação de instâncias estáticas e dinâmicas.

5. (Opcional) Em Configurações adicionais, especifique o seguinte:

a. Opção de compra — selecione instâncias sob demanda, instâncias spot ou um bloco de capacidade existente. Escolha também On-Demand se você planeja usar uma Reserva de Capacidade Sob Demanda (ODCR). Para obter mais informações, consulte Usando ODCRs com o AWS PCS. Escolha Capacity Block para usar uma reserva existente de blocos de capacidade do Amazon EC2 para ML. Para obter mais informações, consulte Usando blocos de capacidade do Amazon EC2 para ML com PCS AWS.

b. Estratégia de alocação — se você selecionou a opção de compra spot, pode especificar como os pools de capacidade spot são escolhidos ao iniciar instâncias no grupo de nós. Para obter mais informações, consulte Estratégias de alocação para instâncias spot no Guia do usuário do Amazon Elastic Compute Cloud. Essa opção não tem efeito se você tiver selecionado a opção de compra sob demanda.

6. (Opcional) Na seção de configurações Slurm personalizadas, você pode adicionar pares de nome e valor do parâmetro para definir configurações adicionais do Slurm. Para obter uma lista completa dos parâmetros compatíveis, consulteConfigurações personalizadas do Slurm para grupos de nós de computação AWS PCS.
7. (Opcional) Em Tags, adicione qualquer tag ao seu grupo de nós de computação.
8. Escolha Criar grupo de nós de computação. O campo Status mostra `Creating` enquanto o AWS PCS provisiona o grupo de nós. Isso pode demorar vários minutos.

Próxima etapa recomendada

- Adicione seu grupo de nós a uma fila no AWS PCS para permitir que ele processe trabalhos.

AWS CLI

Para criar seu grupo de nós de computação usando AWS CLI

Crie sua fila com o comando a seguir. Antes da execução do comando, realize as seguintes substituições:

1. *`region`*Substitua pelo ID do Região da AWS para criar seu cluster, como`us-east-1`.
2. *`my-cluster`*Substitua pelo nome ou pelo nome `clusterId` do seu cluster.
3. *`my-node-group`*Substitua pelo nome do seu grupo de nós de computação. O nome só pode conter caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um caractere alfabético e não pode ter mais de 25 caracteres. O nome deve ser exclusivo dentro do cluster.
4. *`subnet-ExampleID1`*Substitua por uma ou mais sub-redes IDs do seu cluster VPC.
5. *`lt-ExampleID1`*Substitua pelo ID do seu modelo de lançamento personalizado. Se você não tiver um preparado, veja Usando modelos de lançamento do Amazon EC2 com PCS AWS para aprender como criar um.

> **Important**
>
> AWS O PCS cria um modelo de lançamento gerenciado para cada grupo de nós de computação. Esses são nomeadospcs-*`identifier`*-do-not-delete. Não os selecione ao criar ou atualizar um grupo de nós de computação, ou o grupo de nós não funcionará corretamente.

6. *launch-template-version*Substitua por uma versão específica do modelo de lançamento. AWS O PCS associa seu grupo de nós a essa versão específica do modelo de lançamento.
7. *arn:InstanceProfile*Substitua pelo ARN do seu perfil de instância do IAM. Se você não tiver um preparado, consulte Usando modelos de lançamento do Amazon EC2 com PCS AWS para obter orientação.
8. *min-instances*Substitua e *max-instances* por valores inteiros. Você pode definir uma configuração estática, na qual há um número fixo de nós em execução, ou uma configuração dinâmica, na qual até a contagem máxima de nós pode ser executada. Para uma configuração estática, defina o mínimo e o máximo para o mesmo número, maior que zero. Para uma configuração dinâmica, defina o mínimo de instâncias como zero e o máximo de instâncias como um número maior que zero. AWS O PCS não oferece suporte a grupos de nós de computação com uma combinação de instâncias estáticas e dinâmicas.
9. *t3.large*Substitua por outro tipo de instância. Você pode adicionar mais tipos de instância especificando uma lista de `instanceType` configurações. Por exemplo, *.--instance-configs instanceType=c6i.16xlarge instanceType=c6a.16xlarge* Todos os tipos de instância devem ter a mesma arquitetura de processador (x86_64 ou arm64) e número de v. CPUs Se as instâncias tiverem GPUs, todos os tipos de instância deverão ter o mesmo número de GPUs.

```
aws pcs create-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-name my-node-group \
    --subnet-ids subnet-ExampleID1 \
    --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
    --iam-instance-profile-arn=arn:InstanceProfile \
    --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
    --instance-configs instanceType=t3.large
```

Example— Criação de um grupo de nós de computação com configurações personalizadas do Slurm

```
aws pcs create-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-name my-node-group \
    --subnet-ids subnet-ExampleID1 \
    --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
```

```
    --iam-instance-profile-arn=arn:InstanceProfile \
    --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
    --instance-configs instanceType=t3.large \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

Para obter mais informações, consulte Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS.

Há várias configurações opcionais que você pode adicionar ao `create-compute-node-group` comando.

- Você pode especificar `--amiId` se seu modelo de lançamento personalizado não inclui uma referência a uma AMI ou se você deseja substituir esse valor. Observe que a AMI usada para o grupo de nós deve ser compatível com o AWS PCS. Você também pode selecionar uma amostra de AMI fornecida por AWS. Para obter mais informações sobre esse tópico, consulteAmazon Machine Images (AMIs) para AWS PCS.
- Use `--purchase-option` para escolher a forma como o AWS PCS compra instâncias do EC2 para seu grupo de nós de computação. On-Demand é o padrão.
  - `ONDEMAND`— Use instâncias sob demanda. Escolha também essa opção se você planeja usar uma reserva de capacidade sob demanda (ODCR). Para obter mais informações, consulte Usando ODCRs com o AWS PCS.
  - `SPOT`— Use instâncias spot. Se você escolher instâncias spot, também poderá usar `--allocation-strategy` para definir como o AWS PCS escolhe os pools de capacidade spot ao iniciar instâncias no grupo de nós. Para obter mais informações, consulte Estratégias de alocação para instâncias spot no Guia do usuário do Amazon Elastic Compute Cloud.
  - `CAPACITY_BLOCK`— Use um bloco de capacidade existente do Amazon EC2 para reserva de ML. Para obter mais informações, consulte Usando blocos de capacidade do Amazon EC2 para ML com PCS AWS.
- É possível fornecer opções de Slurm configuração para os nós no grupo de nós usando`--slurm-configuration`. Você pode definir o peso (prioridade de agendamento) e a memória real. Os nós com pesos mais baixos têm maior prioridade e as unidades são arbitrárias. Para obter mais informações, consulte Peso na Slurm documentação. A memória real define o tamanho (em GB) da memória real nos nós do grupo de nós. Ele deve ser usado em conjunto com a `CR_CPU_Memory` opção do cluster no AWS PCS em sua Slurm configuração. Para obter mais informações, consulte a RealMemory documentação do Slurm.

> **Important**
>
> A criação do grupo de nós de computação pode levar vários minutos.

Você pode consultar o status do seu grupo de nós com o comando a seguir. Você não poderá associar o grupo de nós a uma fila até que seu status chegue`ACTIVE`.

```
aws pcs get-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group
```

Atualização de um grupo de nós de computação AWS PCS

Este tópico fornece uma visão geral das opções disponíveis e descreve o que considerar ao atualizar um grupo de nós computacionais do AWS PCS. Para obter informações sobre as configurações personalizadas do Slurm, consulte. Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS

Opções para atualizar um grupo de nós computacionais do AWS PCS

A atualização de um grupo de nós computacionais do AWS PCS permite que você altere as propriedades das instâncias lançadas pelo AWS PCS, bem como as regras de como essas instâncias são lançadas. Por exemplo, você pode substituir a AMI para instâncias de grupos de nós por outra com software diferente instalado nela. Ou você pode atualizar os grupos de segurança para alterar a conectividade de rede de entrada ou saída. Você também pode alterar a configuração de escalabilidade e a opção de compra preferida.

As seguintes configurações do grupo de nós não podem ser alteradas após a criação:

- Nome
- Instâncias

Considerações ao atualizar um grupo de nós de computação AWS PCS

Os grupos de nós de computação definem instâncias do EC2 que são usadas para processar trabalhos, fornecer acesso interativo ao shell e outras tarefas. Eles geralmente são associados

a uma ou mais filas AWS PCS. Ao atualizar seu grupo de nós de computação para alterar seu comportamento (ou o de seus nós), considere o seguinte:

- As alterações nas propriedades do grupo de nós de computação entram em vigor quando o status do grupo de nós de computação muda de Atualizando para Ativo. Novas instâncias são lançadas com as propriedades atualizadas.
- As atualizações que não afetam a configuração de nós específicos não afetam os nós em execução. Por exemplo, adicionar uma sub-rede e alterar a estratégia de alocação.
- Se você atualizar o modelo de execução de um grupo de nós de computação, deverá atualizar o grupo de nós de computação para usar a nova versão.
- Para adicionar ou remover um grupo de segurança dos nós em um grupo de nós de computação, edite seu modelo de execução e atualize o grupo de nós de computação. Novas instâncias são lançadas com o conjunto atualizado de grupos de segurança.
- Se você editar diretamente um grupo de segurança usado por um grupo de nós de computação, ele terá efeito imediato nas instâncias em execução e no futuro.
- Se você adicionar ou remover permissões do perfil de instância do IAM usado por um grupo de nós de computação, isso terá efeito imediato nas instâncias em execução e no futuro.
- Para alterar a AMI usada pelas instâncias de um grupo de nós de computação, atualize o grupo de nós de computação (ou seu modelo de execução) para usar a nova AMI e aguarde até que o AWS PCS substitua as instâncias.
- AWS O PCS substitui as instâncias existentes no grupo de nós após uma operação de atualização do grupo de nós. Se houver trabalhos em execução em um nó, esses trabalhos poderão ser concluídos antes que o AWS PCS substitua o nó. Os processos interativos do usuário (como em instâncias de nós de login) são encerrados. O status do grupo de nós retorna para `Active` quando o AWS PCS marca as instâncias para substituição, mas a substituição real ocorre quando as instâncias estão ociosas.
- Se você diminuir o número máximo de instâncias permitido em um grupo de nós de computação, o AWS PCS removerá os nós do Slurm para atingir o novo máximo. AWS O PCS encerra as instâncias em execução associadas aos nós do Slurm removidos. Os trabalhos em execução nos nós removidos falham e retornam às filas.
- AWS O PCS cria um modelo de lançamento gerenciado para cada grupo de nós de computação. Eles são nomeados`pcs-`*`identifier`*`-do-not-delete`. Não os selecione ao criar ou atualizar um grupo de nós de computação, ou o grupo de nós não funcionará corretamente.

- Se você atualizar um grupo de nós de computação para usar o Spot como opção de compra, deverá ter a função vinculada ao serviço AWSServiceRoleForEC2Spot em sua conta. Para obter mais informações, consulte Função spot do Amazon EC2 para PCS AWS.

Para atualizar um grupo de nós computacionais do AWS PCS

Você pode atualizar um grupo de nós usando o AWS Management Console ou o AWS CLI.

Console de gerenciamento da AWS

Para atualizar um grupo de nós de computação

1. Abra o console do AWS PCS em `https://console.aws.amazon.com/pcs/home#/clusters`
2. Selecione o cluster em que você deseja atualizar um grupo de nós de computação.
3. Navegue até os grupos de nós de computação, vá até o grupo de nós que você deseja atualizar e selecione Editar.
4. Nas seções Configuração de computação, Configurações adicionais e Configurações de Slurmpersonalização, atualize todos os valores, exceto:

   - Instâncias — você não pode alterar as instâncias em um grupo de nós de computação.

   Para obter mais informações sobre as configurações personalizadas do Slurm, consulte. Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS
5. Selecione Atualizar. O campo Status mostrará Atualizando enquanto as alterações estão sendo aplicadas.

> **Important**
>
> As atualizações do grupo de nós de computação podem levar vários minutos.

AWS CLI

Para atualizar um grupo de nós de computação

1. Atualize seu grupo de nós de computação com o comando a seguir. Antes da execução do comando, realize as seguintes substituições:

   a. *region-code*Substitua pela região da AWS na qual você deseja criar seu cluster.

   b. *my-node-group*Substitua pelo nome ou `computeNodeGroupId` pelo seu grupo de nós de computação.

   c. *my-cluster*Substitua pelo nome ou pelo nome `clusterId` do seu cluster.

```
aws pcs update-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group
```

   Example— Atualização de um grupo de nós de computação com configurações personalizadas do Slurm

```
aws pcs update-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

   Para obter mais informações, consulte Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS.

2. Atualize todos os parâmetros do grupo de nós, exceto `--instance-configs` o. Por exemplo, para definir um novo ID de AMI, `--amiId my-custom-ami-id` informe onde *my-custom-ami-id* é substituído pela AMI de sua escolha.

> **Important**
>
> A atualização do grupo de nós de computação pode levar vários minutos.

Você pode consultar o status do seu grupo de nós com o comando a seguir.

```
aws pcs get-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group
```

Excluindo um grupo de nós de computação no PCS AWS

Este tópico fornece uma visão geral das opções disponíveis e descreve o que considerar ao excluir um grupo de nós de computação no AWS PCS.

Considerações ao excluir um grupo de nós de computação

Os grupos de nós de computação definem instâncias do EC2 que são usadas para processar trabalhos, fornecer acesso interativo ao shell e outras tarefas. Eles geralmente são associados a uma ou mais filas AWS PCS. Antes de excluir um grupo de nós de computação, considere o seguinte:

- Todas as instâncias do EC2 iniciadas pelo grupo de nós de computação serão encerradas. Isso cancelará os trabalhos que estão sendo executados nessas instâncias e encerrará a execução de processos interativos.
- Você deve desassociar o grupo de nós de computação de todas as filas antes de excluí-lo. Para obter mais informações, consulte Atualizando uma fila AWS PCS.

Excluir o grupo de nós de computação

Você pode usar o Console de gerenciamento da AWS ou AWS CLI para excluir um grupo de nós de computação.

Console de gerenciamento da AWS

Para excluir um grupo de nós de computação

1. Abra o console AWS PCS.
2. Selecione o cluster do grupo de nós de computação.
3. Navegue até grupos de nós de computação e selecione o grupo de nós de computação a ser excluído.
4. Escolha Excluir.

5. O campo Status é exibido`Deleting`. Pode demorar vários minutos para isso ser concluído.

> **Note**
>
> Você pode usar comandos nativos do seu agendador para confirmar se o grupo de nós de computação foi excluído. Por exemplo, use `sinfo` ou `squeue` para o Slurm.

AWS CLI

Para excluir um grupo de nós de computação

- Use o comando a seguir para excluir um grupo de nós de computação com essas substituições:
  - *`region-code`*Substitua por aquele em que Região da AWS seu cluster está.
  - *`my-node-group`*Substitua pelo nome ou ID do seu grupo de nós de computação.
  - *`my-cluster`*Substitua pelo nome ou ID do seu cluster.

```
aws pcs delete-compute-node-group --region region-code \
      --compute-node-group-identifier my-node-group \
      --cluster-identifier my-cluster
```

A exclusão do grupo de nós de computação pode levar vários minutos.

> **Note**
>
> Você pode usar comandos nativos do seu agendador para confirmar se o grupo de nós de computação foi excluído. Por exemplo, use `sinfo` ou `squeue` para o Slurm.

Obtenha detalhes do grupo de nós de computação no AWS PCS

Você pode usar o Console de gerenciamento da AWS or AWS CLI para obter detalhes sobre um grupo de nós de computação, como o ID do grupo de nós de computação, o Amazon Resource Name (ARN) e o ID da Amazon Machine Image (AMI). Esses detalhes geralmente são valores obrigatórios para ações e configurações da API AWS PCS.

Console de gerenciamento da AWS

Para obter detalhes do grupo de nós de computação

1. Abra o console AWS PCS.
2. Selecione o cluster.
3. Escolha grupos de nós de computação.
4. Escolha um grupo de nós de computação no painel da lista.

AWS CLI

Para obter detalhes do grupo de nós de computação

1. Use a ação ListClustersda API para encontrar o nome ou ID do seu cluster.

```
aws pcs list-clusters
```

Exemplos de resultado:

```
{
    "clusters": [
        {
            "name": "get-started-cfn",
            "id": "pcs_abc1234567",
            "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567",
            "createdAt": "2025-04-01T20:11:22+00:00",
            "modifiedAt": "2025-04-01T20:11:22+00:00",
            "status": "ACTIVE"
        }
    ]
}
```

2. Use a ação ListComputeNodeGroupsda API para listar os grupos de nós de computação em um cluster.

```
aws pcs list-compute-node-groups --cluster-identifier cluster-name-or-id
```

Exemplo de chamada:

```
aws pcs list-compute-node-groups --cluster-identifier get-started-cfn
```

Exemplos de resultado:

```
{
    "computeNodeGroups": [
        {
            "name": "compute-1",
            "id": "pcs_abc123abc1",
            "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
computenodegroup/pcs_abc123abc1",
            "clusterId": "pcs_abc1234567",
            "createdAt": "2025-04-01T20:19:25+00:00",
            "modifiedAt": "2025-04-01T20:19:25+00:00",
            "status": "ACTIVE"
        },
        {
            "name": "login",
            "id": "pcs_abc456abc7",
            "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
computenodegroup/pcs_abc456abc7",
            "clusterId": "pcs_abc1234567",
            "createdAt": "2025-04-01T20:19:31+00:00",
            "modifiedAt": "2025-04-01T20:19:31+00:00",
            "status": "ACTIVE"
        }
    ]
}
```

3. Use a ação GetComputeNodeGroupda API para obter detalhes adicionais de um grupo de nós de computação.

```
aws pcs get-compute-node-group --cluster-identifier cluster-name-or-id --
compute-node-group-identifier compute-node-group-name-or-id
```

Exemplo de chamada:

```
aws pcs get-compute-node-group --cluster-identifier get-started-cfn --compute-
node-group-identifier compute-1
```

Exemplos de resultado:

```
{
    "computeNodeGroup": {
        "name": "compute-1",
        "id": "pcs_abc123abc1",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
computenodegroup/pcs_abc123abc1",
        "clusterId": "pcs_abc1234567",
        "createdAt": "2025-04-01T20:19:25+00:00",
        "modifiedAt": "2025-04-01T20:19:25+00:00",
        "status": "ACTIVE",
        "amiId": "ami-0123456789abcdef0",
        "subnetIds": [
            "subnet-abc012345789abc12"
        ],
        "purchaseOption": "ONDEMAND",
        "customLaunchTemplate": {
            "id": "lt-012345abcdef01234",
            "version": "1"
        },
        "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-profile/
AWSPCS-get-started-cfn-us-east-1",
        "scalingConfiguration": {
            "minInstanceCount": 0,
            "maxInstanceCount": 4
        },
        "instanceConfigs": [
            {
                "instanceType": "c6i.xlarge"
            }
        ]
    }
}
```

Encontrando instâncias de grupos de nós de computação no AWS PCS

Cada grupo de nós de computação do AWS PCS pode iniciar instâncias do EC2 com configurações compartilhadas. Você pode usar tags do EC2 para encontrar instâncias em um grupo de nós de computação no Console de gerenciamento da AWS ou com o. AWS CLI

Console de gerenciamento da AWS

Para encontrar suas instâncias do grupo de nós de computação

1. Abra o console AWS PCS.
2. Selecione o cluster.
3. Escolha grupos de nós de computação.
4. Encontre o ID do grupo de nós de login que você criou.
5. Navegue até o console do EC2 e escolha Instâncias.
6. Pesquise as instâncias com a seguinte tag. *node-group-id*Substitua pelo ID (não pelo nome) do seu grupo de nós de computação.

   ```
   aws:pcs:compute-node-group-id=node-group-id
   ```

7. (Opcional) Você pode alterar o valor do estado da instância no campo de pesquisa para encontrar instâncias que estão sendo configuradas ou que foram encerradas recentemente.
8. Encontre o ID da instância e o endereço IP de cada instância na lista de instâncias marcadas.

AWS CLI

Para encontrar suas instâncias de grupo de nós, use os comandos a seguir. Antes de executar os comandos, faça as seguintes substituições:

- *region-code*Substitua pelo Região da AWS do seu cluster. Exemplo: `us-east-1`
- *node-group-id*Substitua pelo ID (não pelo nome) do seu grupo de nós de computação. Para encontrar a ID de um grupo de nós de computação, consulteObtenha detalhes do grupo de nós de computação no AWS PCS.

- `running`Substitua por outros estados de instância, como `pending` ou `terminated` para encontrar instâncias do EC2 em outros estados.

```
aws ec2 describe-instances \
     --region region-code --filters \
     "Name=tag:aws:pcs:compute-node-group-id,Values=node-group-id" \
     "Name=instance-state-name,Values=running" \
     --query 'Reservations[*].Instances[*].
{InstanceID:InstanceId,State:State.Name,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress}
```

Esse comando retorna uma saída semelhante à seguinte. O valor de `PublicIP` é `null` se a instância estiver em uma sub-rede privada.

```
[
    [
        {
            "InstanceID": "i-0123456789abcdefa",
            "State": "running",
            "PublicIP": "18.189.32.188",
            "PrivateIP": "10.0.0.1"
        }
    ]
]
```

> **Note**
>
> Se você espera `describe-instances` retornar um grande número de instâncias, deve usar opções para várias páginas. Para obter mais informações, consulte DescribeInstancesa Amazon Elastic Compute Cloud API Reference.

Usando modelos de lançamento do Amazon EC2 com PCS AWS

No Amazon EC2, um modelo de lançamento pode armazenar um conjunto de preferências para que você não precise especificá-las individualmente ao iniciar instâncias. AWS O PCS incorpora modelos de lançamento como uma forma flexível de configurar grupos de nós de computação. Ao criar um grupo de nós, você fornece um modelo de lançamento. AWS O PCS cria um modelo de lançamento derivado que inclui transformações para ajudar a garantir que ele funcione com o serviço.

Entender quais são as opções e considerações ao escrever um modelo de lançamento personalizado pode ajudá-lo a criar um para uso com o AWS PCS. Para obter mais informações sobre modelos de execução, consulte Launching an Instance from a Launch an instance from a launch template no Amazon EC2 User Guide.

Tópicos

Visão geral dos modelos de lançamento no AWS PCS

Há mais de 30 parâmetros disponíveis que você pode incluir em um modelo de execução do EC2, controlando muitos aspectos de como as instâncias são configuradas. A maioria é totalmente compatível com o AWS PCS, mas há algumas exceções.

Os seguintes parâmetros do modelo EC2 Launch serão ignorados pelo AWS PCS, pois essas propriedades precisam ser gerenciadas diretamente pelo serviço:

- Atributos do tipo de type/Specify instância (`InstanceRequirements`) — O AWS PCS não oferece suporte à seleção de instância baseada em atributos.
- Tipo de instância (`InstanceType`) — Especifique os tipos de instância ao criar um grupo de nós.
- Perfil de details/IAM instância avançado (`IamInstanceProfile`) — Você fornece isso ao criar ou atualizar o grupo de nós.

- Terminação avançada de details/Disable API (`DisableApiTermination`) — O AWS PCS deve controlar o ciclo de vida das instâncias do grupo de nós que ele executa.
- Advanced details/Disable API stop (`DisableApiStop`) — O AWS PCS deve controlar o ciclo de vida das instâncias do grupo de nós que ele executa.
- Avançado details/Stop — Comportamento de hibernação (`HibernationOptions`) — O AWS PCS não suporta hibernação de instâncias.
- details/Elastic GPU avançada (`ElasticGpuSpecifications`) — A Amazon Elastic Graphics chegou ao fim da vida útil em 8 de janeiro de 2024.
- details/Elastic Inferência avançada (`ElasticInferenceAccelerators`) — O Amazon Elastic Inference não está mais disponível para novos clientes.
- AAdvanced details/Specify CPU options/Threadspor núcleo (`ThreadsPerCore`) — O AWS PCS define o número de fios por núcleo como 1.

Esses parâmetros têm requisitos especiais que oferecem suporte à compatibilidade com o AWS PCS:

- Dados do usuário (`UserData`) — Isso deve ser codificado em várias partes. Consulte Trabalhando com dados de usuário do Amazon EC2 para PCS AWS.
- Imagens do aplicativo e do sistema operacional (`ImageId`) — Você pode incluir isso. No entanto, se você especificar uma ID de AMI ao criar ou atualizar o grupo de nós, ela substituirá o valor no modelo de execução. A AMI que você fornece deve ser compatível com o AWS PCS. Para obter mais informações, consulte "Amazon Machine Images (AMIs) para AWS PCS.
- Rede settings/Firewall (grupos de segurança) (`SecurityGroups`) — Uma lista de nomes de grupos de segurança não pode ser definida em um modelo de lançamento do AWS PCS. Você pode definir uma lista de grupos de segurança IDs (`SecurityGroupIds`), a menos que defina interfaces de rede no modelo de execução. Em seguida, você deve especificar o grupo de segurança IDs para cada interface. Para obter mais informações, consulte Grupos de segurança no AWS PCS.
- Configuração de settings/Advanced rede de rede (`NetworkInterfaces`) — Se você usa instâncias do EC2 com uma única placa de rede e não exige nenhuma configuração de rede especializada, o AWS PCS pode configurar a rede de instâncias para você. Para configurar várias placas de rede ou habilitar o Elastic Fabric Adapter em suas instâncias, use`NetworkInterfaces`. Cada interface de rede deve ter uma lista de grupos de segurança IDs abaixo`Groups`. Para obter mais informações, consulte Várias interfaces de rede no AWS PCS.

- Detalhes avançados/reserva de capacidade (`CapacityReservationSpecification`) — Isso pode ser definido, mas não pode fazer referência a um específico `CapacityReservationId` ao trabalhar com AWS o PCS. No entanto, você pode referenciar um grupo de reserva de capacidade, onde esse grupo contém uma ou mais reservas de capacidade. Para obter mais informações, consulte Reservas de capacidade no AWS PCS.

Criar um modelo de execução básico

Você pode criar um modelo de lançamento usando o Console de gerenciamento da AWS ou AWS CLI o.

Console de gerenciamento da AWS

Para criar um modelo de execução

1. Abra o EC2console da Amazon e selecione Modelos de lançamento.
2. Escolha Criar modelo de execução.
3. Em Nome e descrição do modelo do Launch, insira um nome exclusivo e distinto para o nome do modelo do Launch.
4. Em Par de chaves (login) em Nome do par de chaves, selecione o par de chaves SSH que será usado para fazer login em EC2 instâncias gerenciadas pelo AWS PCS. Isso é opcional, mas recomendado.
5. Em Configurações de rede, depois em Firewall (grupos de segurança), escolha grupos de segurança a serem anexados à interface de rede. Todos os grupos de segurança no modelo de execução devem ser do seu cluster AWS PCS VPC. No mínimo, escolha:
   - Um grupo de segurança que permite a comunicação com o cluster AWS PCS
   - Um grupo de segurança que permite a comunicação entre EC2 instâncias iniciadas pelo AWS PCS
   - (Opcional) Um grupo de segurança que permite acesso SSH de entrada a instâncias interativas
   - (Opcional) Um grupo de segurança que permite que os nós de computação façam conexões de saída com a Internet
   - (Opcional) Grupos de segurança que permitem acesso a recursos em rede, como sistemas de arquivos compartilhados ou um servidor de banco de dados.

6. Seu novo ID do modelo de lançamento estará acessível no EC2 console da Amazon em Modelos de lançamento. O ID do modelo de lançamento terá o formulário`lt-0123456789abcdef01`.

Próxima etapa recomendada

- Use o novo modelo de execução para criar ou atualizar um grupo de nós de computação AWS PCS.

AWS CLI

Para criar um modelo de execução

Crie seu modelo de lançamento com o comando a seguir.

- Antes da execução do comando, realize as seguintes substituições:
  a. *region-code*Substitua pelo Região da AWS local em que você está trabalhando com o AWS PCS
  b. *my-launch-template-name*Substitua por um nome para seu modelo. Ele deve ser exclusivo do Conta da AWS e Região da AWS que você está usando.
  c. *my-ssh-key-name*Substitua pelo nome da sua chave SSH preferida.
  d. Substitua *sg-ExampleID1* e *sg-ExampleID2* por um grupo de segurança IDs que permite a comunicação entre suas EC2 instâncias e o agendador e a comunicação entre EC2 instâncias. Se você tiver apenas um grupo de segurança que habilite todo esse tráfego, poderá remover o `sg-ExampleID2` caractere de vírgula anterior. Você também pode adicionar mais grupos de segurança IDs. Todos os grupos de segurança que você inclui no modelo de execução devem ser do seu cluster AWS PCS VPC.

```
aws ec2 create-launch-template --region region-code \
    --launch-template-name my-template-name \
    --launch-template-data '{"KeyName":"my-ssh-key-name","SecurityGroupIds":
 ["sg-ExampleID1","sg-ExampleID2"]}'
```

AWS CLI Isso exibirá um texto semelhante ao seguinte. O ID do modelo de lançamento é encontrado em`LaunchTemplateId`.

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0123456789abcdef01",
        "LaunchTemplateName": "my-launch-template-name",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-04-30T18:16:06.000Z"
    }
}
```

Próxima etapa recomendada

- Use o novo modelo de execução para criar ou atualizar um grupo de nós de computação AWS PCS.

Trabalhando com dados de usuário do Amazon EC2 para PCS AWS

Você pode fornecer dados do usuário do EC2 em seu modelo de execução que `cloud-init` é executado quando suas instâncias são iniciadas. Os blocos de dados do usuário com o tipo de conteúdo são `cloud-config` executados antes do registro da instância na API AWS PCS, enquanto os blocos de dados do usuário com o tipo de conteúdo são `text/x-shellscript` executados após a conclusão do registro, mas antes do início do daemon do Slurm. Para obter mais informações sobre os tipos de conteúdo, consulte a documentação do cloud-init.

nossos dados de usuário podem realizar cenários de configuração comuns, incluindo, mas não se limitando ao seguinte:

- Incluindo usuários ou grupos
- Instalando pacotes
- Criação de partições e sistemas de arquivos
- Montagem de sistemas de arquivos de rede

Os dados do usuário nos modelos de lançamento devem estar no formato de arquivamento de várias partes MIME. Isso ocorre porque seus dados de usuário são mesclados com outros dados de usuário

do AWS PCS que são necessários para configurar nós em seu grupo de nós. É possível combinar vários blocos de dados de usuário em um único arquivo MIME de várias partes.

Um arquivo em várias partes MIME consiste nos seguintes componentes:

- O tipo de conteúdo e a declaração de limite da parte: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- A declaração da versão MIME: `MIME-Version: 1.0`
- Um ou mais blocos de dados do usuário que contêm os seguintes componentes:
  - O limite de abertura, que sinaliza o início de um bloco de dados do usuário: `--==BOUNDARY==`. Você deve manter a linha antes desse limite em branco.
  - A declaração do tipo de conteúdo para o bloco: `Content-Type: text/cloud-config; charset="us-ascii"` ou`Content-Type: text/x-shellscript; charset="us-ascii"`. Você deve manter a linha após o branco da declaração do tipo de conteúdo.
  - O conteúdo de dados do usuário, por exemplo, uma lista de comandos de shell ou diretivas do `cloud-config`.
- O limite de fechamento que sinaliza o fim do arquivo MIME de várias partes: `--==BOUNDARY==--`. Você deve manter a linha antes do branco do limite de fechamento.

> **Note**
>
> Se você adicionar dados do usuário a um modelo de lançamento no console do Amazon EC2, poderá colá-los como texto sem formatação. Ou você pode fazer o upload de um arquivo. Se você usa o AWS CLI ou um AWS SDK, deve primeiro codificar em base64 os dados do usuário e enviar essa string como o valor do `UserData` parâmetro ao chamar CreateLaunchTemplate, conforme mostrado neste arquivo JSON.

```
{
    "LaunchTemplateName": "base64-user-data",
    "LaunchTemplateData": {
        "UserData":
 "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
    }
}
```

Exemplos

- Exemplo: instalar software a partir de um repositório de pacotes
- Exemplo: executar scripts a partir de um bucket do S3
- Exemplo: definir variáveis de ambiente globais
- Usando sistemas de arquivos de rede com AWS PCS
- Exemplo: usar um sistema de arquivos EFS como um diretório inicial compartilhado

Exemplo: instalar software para AWS PCS a partir de um repositório de pacotes

Forneça esse script como valor de `"userData"` em seu modelo de lançamento. Para obter mais informações, consulte Trabalhando com dados de usuário do Amazon EC2 para PCS AWS.

Esse script usa cloud-config para instalar pacotes de software em instâncias de grupos de nós no lançamento. Para obter mais informações, consulte os formatos de dados do usuário na documentação do cloud-init. Este exemplo instala `curl` e. `llvm`

> **Note**
>
> Suas instâncias devem ser capazes de se conectar aos repositórios de pacotes configurados.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- python3-devel
- rust
- golang

--==MYBOUNDARY==--
```

Exemplo: executar scripts adicionais para AWS PCS a partir de um bucket do S3

Forneça esse script como valor de `"userData"` em seu modelo de lançamento. Para obter mais informações, consulte Trabalhando com dados de usuário do Amazon EC2 para PCS AWS.

O script de dados do usuário a seguir usa cloud-config para importar um script de um bucket do S3 e executá-lo em instâncias de grupos de nós na inicialização. Para obter mais informações, consulte os formatos de dados do usuário na documentação do cloud-init.

Substitua os valores a seguir pelos seus próprios detalhes:

- *amzn-s3-demo-bucket*— O nome de um bucket do S3 que sua conta pode ler.
- *object-key*— A chave do objeto S3 do script a ser importado. Isso inclui o nome do script e sua localização na estrutura de pastas do bucket. Por exemplo, `.scripts/script.sh` Para obter mais informações, consulte Organização de objetos no console do Amazon S3 usando pastas no Guia do usuário do Amazon Simple Storage Service.
- *shell*— O shell Linux a ser usado para executar o script, como`bash`.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/object-key /tmp/script.sh
- /usr/bin/shell /tmp/script.sh

--==MYBOUNDARY==--
```

O perfil da instância do IAM para o grupo de nós deve ter acesso ao bucket. A política do IAM a seguir é um exemplo do bucket no script de dados do usuário acima.

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Exemplo: definir variáveis de ambiente globais para AWS PCS

Forneça esse script como o valor de `"userData"` em seu modelo de lançamento. Para obter mais informações, consulte Trabalhando com dados de usuário do Amazon EC2 para PCS AWS.

O exemplo a seguir é usado `/etc/profile.d` para definir variáveis globais em instâncias de grupos de nós.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
touch /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR1=100 >> /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR2=abc >> /etc/profile.d/awspcs-userdata-vars.sh

--==MYBOUNDARY==--
```

Exemplo: Use um sistema de arquivos EFS como um diretório inicial compartilhado para AWS PCS

Forneça esse script como valor de `"userData"` em seu modelo de lançamento. Para obter mais informações, consulte Trabalhando com dados de usuário do Amazon EC2 para PCS AWS.

Este exemplo estende o exemplo de montagem do EFS Usando sistemas de arquivos de rede com AWS PCS para implementar um diretório inicial compartilhado. O conteúdo de /home é copiado antes da montagem do sistema de arquivos EFS. O conteúdo é então copiado rapidamente para o armazenamento compartilhado após a conclusão da montagem.

Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *`/mount-point-directory`*— O caminho em uma instância em que você deseja montar o sistema de arquivos EFS.
- *`filesystem-id`*— O ID do sistema de arquivos do sistema de arquivos EFS.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /tmp/home
  - rsync -a /home/ /tmp/home
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults
  - rsync -a --ignore-existing /tmp/home/ /home
  - rm -rf /tmp/home/

--==MYBOUNDARY==--
```

Exemplo: habilitar o SSH sem senha

Você pode usar o exemplo do diretório inicial compartilhado para implementar conexões SSH entre instâncias de cluster usando chaves SSH. Para cada usuário que usa o sistema de arquivos inicial compartilhado, execute um script semelhante ao seguinte:

```
#!/bin/bash

mkdir -p $HOME/.ssh && chmod 700 $HOME/.ssh
touch $HOME/.ssh/authorized_keys
chmod 600 $HOME/.ssh/authorized_keys

if [ ! -f "$HOME/.ssh/id_rsa" ]; then
    ssh-keygen -t rsa -b 4096 -f $HOME/.ssh/id_rsa -N ""
    cat ~/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
fi
```

> **Note**
>
> As instâncias devem usar um grupo de segurança que permita conexões SSH entre os nós do cluster.

Reservas de capacidade no AWS PCS

Você pode reservar a EC2 capacidade da Amazon em uma zona de disponibilidade específica e por um período específico usando reservas de capacidade sob demanda ou blocos de EC2 capacidade da Amazon para ML para garantir que você tenha a capacidade computacional necessária disponível quando precisar.

As reservas de capacidade sob demanda (ODCRs) permitem que você reserve capacidade computacional para suas EC2 instâncias da Amazon em uma zona de disponibilidade específica por qualquer período. Você pode criar e cancelar reservas a qualquer momento, sem compromissos de longo prazo ou pagamentos antecipados. ODCRs são ideais quando você precisa de reservas de capacidade flexíveis que podem ser modificadas conforme suas necessidades mudam. Para obter mais informações, consulte Reservas de capacidade sob demanda no Guia do usuário do Amazon Elastic Compute Cloud.

O Amazon EC2 Capacity Blocks for ML permite que você reserve instâncias de computação acelerada baseadas em GPU para uso futuro, com até 8 semanas de antecedência. Você pode

reservar blocos de 1 a 64 instâncias por períodos de 1 dia a 6 meses. Os blocos de capacidade são ideais para cargas de trabalho de aprendizado de máquina que exigem acesso garantido à capacidade da GPU em horários específicos. Para obter mais informações, consulte Capacity Blocks for ML no Guia do usuário do Amazon Elastic Compute Cloud.

Tópicos

Usando ODCRs com o AWS PCS

Você pode escolher como o AWS PCS consome suas instâncias reservadas. Se você criar um ODCR aberto, todas as instâncias correspondentes iniciadas pelo AWS PCS ou outros processos em sua conta serão contabilizadas na reserva. Com um ODCR direcionado, somente as instâncias iniciadas com o ID de reserva específico são contabilizadas na reserva. Para cargas de trabalho urgentes, as segmentações ODCRs são mais comuns.

Você pode configurar um grupo de nós de computação AWS PCS para usar um ODCR direcionado adicionando-o a um modelo de execução. Aqui estão as etapas para fazer isso:

1. Crie uma reserva de capacidade sob demanda (ODCR) direcionada usando o Guia do usuário do Amazon EC2 Create a Capacity Reservation.
2. Associe o ODCR a um modelo de lançamento. Há duas maneiras de fazer isso:

    a. Associação direta do ODCR: faça referência ao ID do ODCR diretamente no modelo de lançamento. Essa abordagem fornece controle estrito de capacidade e não oferece suporte ao preenchimento de instâncias (se o grupo de nós de computação solicitar mais instâncias do que as disponíveis no ODCR, nenhuma instância adicional será iniciada).

    b. Associação ao grupo de reserva de capacidade: adicione o ODCR a um grupo de reserva de capacidade e faça referência ao grupo no modelo de lançamento. Essa abordagem oferece suporte ao preenchimento de instâncias, permitindo que a AWS PCS lance instâncias sob demanda adicionais se a capacidade de reserva for excedida.
3. Crie ou atualize um grupo de nós de computação AWS PCS para usar o modelo de lançamento. Para obter mais informações, consulte o Guia do usuário do AWS PCS Compute Node Groups.

    - Defina o `purchaseOption` do grupo de nós de computação como `ONDEMAND`.

Exemplo: reserve e use instâncias hpc6a.48xlarge com um ODCR direcionado

Esse exemplo de comando cria um ODCR direcionado para 32 instâncias hpc6a.48xlarge. Para iniciar as instâncias reservadas em um grupo de posicionamento, adicione `--placement-group-arn` ao comando. Você pode definir uma data de parada com `--end-date` e`--end-date-type`, caso contrário, a reserva continuará até que seja encerrada manualmente.

```
aws ec2 create-capacity-reservation \
    --instance-type hpc6a.48xlarge \
    --instance-platform Linux/UNIX \
    --availability-zone us-east-2a \
    --instance-count 32 \
    --instance-match-criteria targeted
```

O resultado desse comando será um ARN para o novo ODCR. O ID do ODCR pode ser recuperado do ARN `"arn:aws:ec2:us-east-2:123456789012:capacity-reservation/ODCR-ID"` ou usando o Amazon EC2. DescribeCapacityReservations

Associação direta do ODCR: adicione o ID do ODCR ao modelo de lançamento. Aqui está um exemplo de modelo de lançamento que faz referência à ID do ODCR.

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationTarget": {
      "CapacityReservationId": "cr-1234567890abcdef1"
    }
  }
}
```

Associação ao grupo de reserva de capacidade: crie um grupo de reserva de capacidade e adicione o grupo ao modelo de lançamento. O comando a seguir cria um grupo de reserva de capacidade chamado`EXAMPLE-CR-GROUP`.

```
aws resource-groups create-group \
    --name EXAMPLE-CR-GROUP \
    --configuration \
        '{"Type": "AWS::EC2::CapacityReservationPool"}' \
        '{"Type": "AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

O comando a seguir adiciona o ODCR ao grupo de reserva de capacidade.

```
aws resource-groups group-resources --group EXAMPLE-CR-GROUP \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/
cr-1234567890abcdef1
```

Com o ODCR criado e adicionado a um grupo de reserva de capacidade, agora ele pode ser conectado a um grupo de nós de computação do AWS PCS adicionando-o a um modelo de execução. Aqui está um exemplo de modelo de lançamento que faz referência ao grupo de reserva de capacidade.

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-2:123456789012:group/EXAMPLE-CR-GROUP"
  }
}
```

Por fim, crie ou atualize um grupo de nós de computação AWS PCS para usar instâncias hpc6a.48xlarge e use o modelo de execução que faz referência ao ODCR. Para um grupo de nós estático, defina instâncias mínima e máxima para o tamanho da reserva (32). Para um grupo dinâmico de nós, defina o mínimo de instâncias como 0 e o máximo para o tamanho de instância desejado.

Este exemplo é uma implementação simples de um único ODCR provisionado para um grupo de nós de computação. Mas, o AWS PCS suporta muitos outros designs. Por exemplo, você pode subdividir um grande grupo ODCR ou de reserva de capacidade entre vários grupos de nós de computação. Ou você pode usar ODCRs aquela outra conta da AWS criada e compartilhada com a sua.

Para obter mais informações, consulte Reservas de capacidade sob demanda e blocos de capacidade para ML no Guia do usuário do Amazon Elastic Compute Cloud.

Usando blocos de capacidade do Amazon EC2 para ML com PCS AWS

O Amazon EC2 Capacity Blocks for ML é uma opção de compra do Amazon EC2 que permite que você pague antecipadamente para reservar instâncias de computação acelerada baseadas em GPU dentro de um intervalo específico de data e hora para suportar cargas de trabalho de curta duração. As instâncias que são executadas dentro de um bloco de capacidade são automaticamente colocadas próximas umas das outras dentro do Amazon EC2 UltraClusters, para redes de baixa

latência, em escala de petabits e sem bloqueio. Para obter mais informações, consulte Capacity Blocks for ML no Guia do usuário do Amazon Elastic Compute Cloud.

Você pode usar um modelo de execução para que o AWS PCS use um bloco de capacidade ao iniciar instâncias para um grupo de nós de computação.

> **Note**
>
> AWS O PCS introduziu suporte para blocos de capacidade desde a versão 24.05 do Slurm.

Limitações

- AWS O PCS suporta somente blocos de capacidade com famílias de instâncias P5en, P5e, P5 e P4d.
- Você só pode associar um grupo de nós de computação a 1 bloco de capacidade por vez.
- Você não pode associar um grupo de nós de computação a um grupo de reserva de capacidade que combine vários blocos de capacidade.
- Os blocos de capacidade devem estar em um `active` estado `scheduled` ou para serem usados com o AWS PCS. Você não pode usar blocos de capacidade em outros estados, como`payment-failed`. Para obter mais informações, consulte Exibir blocos de capacidade no Guia do usuário do Amazon Elastic Compute Cloud.

Expiração do bloco de capacidade

Os blocos de capacidade são limitados a um intervalo específico de data e hora. Quando um bloco de capacidade expira:

- O grupo de nós de computação associado a esse bloco de capacidade continua existindo e permanece associado às mesmas filas.
- Todas as instâncias no grupo de nós de computação são encerradas e os trabalhos ativos podem falhar, com base nas configurações do Slurm.
- AWS O PCS não pode iniciar novas instâncias no grupo de nós de computação.
- Todos os trabalhos em fila ou recém-enviados permanecem em estado pendente até que outro grupo de nós de computação seja anexado à fila ou você atualize o grupo de nós de computação para usar um novo modelo de execução que especifique um novo bloco de capacidade.

Configurar um grupo de nós de computação AWS PCS para usar um bloco de capacidade

Para associar um bloco de capacidade a um grupo de nós de computação

1. Crie um modelo de EC2 lançamento da Amazon para AWS PCS que especifique seu bloco de capacidade. Para obter mais informações sobre a criação de um modelo de lançamento para AWS PCS, consulteUsando modelos de lançamento do Amazon EC2 com PCS AWS.

   Seu modelo de lançamento deve incluir:

   - O valor `MarketType` de `InstanceMarketOptions` deve ser definido como`capacity-block`.
   - A `CapacityReservationSpecification` com um válido `CapacityReservationId`
   - Um válido `InstanceType` que corresponda ao tipo de instância do bloco de capacidade que você comprou.
2. Crie um grupo de nós de computação que use o modelo de execução. Para obter mais informações, consulte Criação de um grupo de nós de computação no AWS PCS. Você também pode atualizar um grupo de nós de computação existente para usar o modelo de execução. Para obter mais informações, consulte Atualização de um grupo de nós de computação AWS PCS.

   Quando você cria ou atualiza o grupo de nós de computação:

   - A identidade do IAM que você usa para criar ou atualizar o grupo de nós de computação precisa ter a seguinte permissão:

     ```
     ec2:DescribeCapacityReservations
     ```

     Para obter mais informações, consulte Permissões mínimas para AWS PCS.
   - O bloco de capacidade deve estar em um `active` estado `scheduled` ou.
   - Defina o `purchaseOption` do grupo de nós de computação como`CAPACITY_BLOCK`.
   - O `maxInstanceCount` do grupo de nós de computação não deve exceder o tamanho do bloco de capacidade.
   - A zona de disponibilidade do grupo de nós de computação deve corresponder a 1 das zonas de disponibilidade de sub-rede do grupo de nós de computação.

> **Important**
>
> Você não pode alterar o tipo de instância de um grupo de nós de computação ao atualizá-lo. Você só pode usar um bloco de capacidade com o mesmo tipo de instância do grupo de nós de computação. Se quiser usar um bloco de capacidade com um tipo de instância diferente, você deve criar um novo grupo de nós de computação.

Perguntas frequentes sobre o uso de blocos de capacidade com AWS PCS

Acabei de pagar por um bloco de capacidade e imediatamente tentei usá-lo com o AWS PCS, mas a criação do grupo de nós de computação falhou. O que aconteceu?

Seu bloco de capacidade pode não estar em um `active` estado `scheduled` ou. Tente novamente depois que o bloco de capacidade for `scheduled` ou`active`.

Estou usando um bloco de capacidade no AWS PCS e comprei uma extensão antes que ela expirasse. Como continuo a usá-lo no AWS PCS?

Você não precisa fazer nada para continuar usando o Bloco de Capacidade no AWS PCS. A data de término do seu Bloco de Capacidade é atualizada após o pagamento da extensão ser bem-sucedido. Enquanto seu bloco de capacidade não expirar, o grupo de nós de computação continuará operando. Se o pagamento da extensão falhar, seu Bloco de Capacidade permanecerá `active` e o grupo de nós de computação operará até que o Bloco de Capacidade expire na data de término original.

O que acontece com meus trabalhos em fila e em execução se meu bloco de capacidade expirar?

Os trabalhos em fila que não foram iniciados antes da expiração do Bloco de Capacidade permanecem pendentes até que você anexe outro grupo de nós de computação à fila ou atualize o grupo de nós de computação com um novo Bloco de Capacidade. Você ainda pode enviar trabalhos para a fila. Suas configurações do Slurm afetam os trabalhos ativos. Por padrão, os trabalhos ativos são automaticamente enfileirados novamente, mas podem apresentar erros ou falhar.

Meu bloco de capacidade expirou. Devo fazer alguma coisa?

Você não precisa fazer nada. Você pode verificar o status de suas reservas de capacidade do EC2 no console do Amazon EC2. Quando um bloco de capacidade expira, o grupo de nós de computação associado a esse bloco de capacidade continua existindo e manipulando as mesmas filas. O grupo de nós de computação não tem nenhuma instância para executar trabalhos. Você

pode excluir o grupo de nós de computação ou desassociá-lo das filas para impedir que os usuários enviem trabalhos que não serão executados.

Quero usar um novo bloco de capacidade com meu grupo de nós de computação AWS PCS. O que devo fazer?

Recomendamos que você crie um novo grupo de nós de computação para usar o novo Bloco de Capacidade. Para obter mais informações, consulte Configurar um grupo de nós de computação AWS PCS para usar um bloco de capacidade.

Como posso compartilhar 1 bloco de capacidade entre clusters e serviços?

Você pode dividir um bloco de capacidade em vários clusters e serviços. Por exemplo, para dividir um bloco de capacidade com 64 `p5.48xlarge` instâncias com 20 nós no PCS-Cluster-1, 16 nós no PCS-Cluster-2 e os nós restantes para outros serviços, defina os dois como 20 no PCS-Cluster-1 `minInstanceCount` e `maxInstanceCount` 16 no PCS-Cluster-2.

Posso usar mais de 1 bloco de capacidade ou capacidade combinada com 1 grupo de nós de computação?

Não. Somente 1 bloco de capacidade pode ser associado a um único grupo de nós de computação. AWS O PCS não suporta grupos de reserva de capacidade que combinam vários blocos de capacidade.

Como sei quando meus blocos de capacidade começam ou expiram?

Independentemente do AWS PCS, o Amazon EC2 envia um `Capacity Block Reservation Delivered` evento EventBridge quando uma reserva do Bloco de Capacidade começa e um `Capacity Block Reservation Expiration Warning` evento 40 minutos antes da expiração da reserva do Bloco de Capacidade. Para obter mais informações, consulte Monitorar blocos de capacidade usando EventBridge o Guia do usuário do Amazon Elastic Compute Cloud.

Como o Slurm rastreia o estado do meu bloco de capacidade?

Você pode executar `sinfo` para entender como o AWS PCS usa o Bloco de Capacidade. No exemplo de saída a seguir, uma fila está associada a um grupo de nós de computação que executa 4 instâncias de um bloco de `active` capacidade. Os nós estão no estado `idle` Slurm (disponíveis para uso e ainda não estão alocados para nenhuma tarefa).

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
fanout up infinite 4 idle node-fanout-[1-4]
```

Se, em vez disso, os nós estiverem no `maint` estado, você poderá executar `scontrol show res` para ver detalhes sobre a reserva do Slurm que controla esse estado. No exemplo de saída a seguir, o Bloco de Capacidade está `scheduled` com uma data de início futura.

```
$ scontrol show res

ReservationName=node-fanout-scheduled StartTime=2025-10-14T13:09:17
 EndTime=2025-10-14T13:11:17 Duration=00:02:00
   Nodes=node-fanout-[1-4] NodeCnt=4 CoreCnt=16 Features=(null) PartitionName=(null)
 Flags=MAINT,SPEC_NODES
   TRES=cpu=16

   Users=root Groups=(null) Accounts=(null) Licenses=(null) State=ACTIVE
 BurstBuffer=(null)
   MaxStartDelay=(null)

   Comment=node-fanout Scheduled
```

Como posso saber se os erros que estou recebendo ao iniciar a capacidade são porque meu bloco de capacidade está compartilhado?

Verifique as reservas de capacidade no console do Amazon EC2 para descobrir quantas instâncias do bloco de capacidade estão ativamente provisionadas. Verifique as tags de cada instância para descobrir qual serviço ou cluster a usa. Por exemplo, todas as instâncias do AWS PCS têm tags AWS PCS, como as `aws:pcs:cluster-id = pcs_l0mizqyk5o | aws:pcs:compute-node-group-id = pcs_ic7onkmfqk` que indicam a quais clusters e grupos de nós de computação a instância pertence. Em seguida, você pode verificar se o bloco de capacidade está na capacidade máxima.

Você usa `scontrol show nodes` para verificar se um nó do Capacity Block em um cluster AWS PCS está acionando`ReservationCapacityExceeded`:

```
[root@ip-172-16-10-54 ~]# scontrol show nodes test-node-8-gamma-cb-2
NodeName=test-8-gamma-cb-2 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=8 CPUTot=8 CPULoad=0.00
   AvailableFeatures=test-8-gamma-cb,gpu
   ActiveFeatures=test-8-gamma-cb,gpu
   Gres=gpu:H100:1
   NodeAddr=test-8-gamma-cb-2 NodeHostName=test-8-gamma-cb-2
   RealMemory=249036 AllocMem=0 FreeMem=N/A Sockets=8 Boards=1
```

```
   State=IDLE+CLOUD+POWERING_DOWN ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A
 MCS_label=N/A
   Partitions=my-q
   BootTime=None SlurmdStartTime=None
   LastBusyTime=Unknown ResumeAfterTime=None
   CfgTRES=cpu=8,mem=249036M,billing=8
   AllocTRES=
   CurrentWatts=0 AveWatts=0
   Reason=Failed to launch backing instance (Error Code:
 ReservationCapacityExceeded) [root@2025-08-28T15:15:33]
```

Quando vários grupos de nós de computação estão conectados à mesma fila, como posso forçar a execução de um trabalho em instâncias com suporte do Capacity Block?

Você pode usar os recursos e restrições do Slurm para bloquear uma tarefa em um determinado conjunto de nós. Recomendamos que você não defina pesos do Slurm para cada grupo de nós de computação, pois isso só funciona com nós que não estão no estado. `maint`

Parâmetros úteis do modelo de lançamento

Esta seção descreve alguns parâmetros do modelo de execução que podem ser amplamente úteis com o AWS PCS.

Ativar o CloudWatch monitoramento detalhado

Você pode ativar a coleta de CloudWatch métricas em um intervalo menor usando um parâmetro de modelo de lançamento.

Console de gerenciamento da AWS

Nas páginas do console para criar ou editar modelos de lançamento, essa opção é encontrada na seção Detalhes avançados. Defina CloudWatch Monitoramento detalhado como Ativar.

YAML

```
Monitoring:
    Enabled: True
```

JSON

```
{"Monitoring": {"Enabled": "True"}}
```

Para obter mais informações, consulte Ativar ou desativar o monitoramento detalhado de suas instâncias no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Serviço de metadados de instância versão 2 (IMDS v2)

O uso do IMDS v2 com instâncias do EC2 oferece aprimoramentos significativos de segurança e ajuda a mitigar os riscos potenciais associados ao acesso aos metadados da instância em ambientes. AWS

Console de gerenciamento da AWS

Nas páginas do console para criar ou editar modelos de lançamento, essa opção é encontrada na seção Detalhes avançados. Defina os metadados acessíveis como Ativados, a versão dos metadados somente como V2 (é necessário um token) e o limite de salto de resposta dos metadados como 4.

YAML

```
MetadataOptions:
  HttpEndpoint: enabled
  HttpTokens: required
  HttpPutResponseHopLimit: 4
```

JSON

```
{
    "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpPutResponseHopLimit": 4,
        "HttpTokens": "required"
    }
}
```

.

AWS Filas PCS

Uma fila AWS PCS é uma abstração leve sobre a implementação nativa de uma fila de trabalho do agendador. No caso do Slurm, uma fila AWS PCS é equivalente a uma partição do Slurm.

Os usuários enviam trabalhos para uma fila onde residem até que possam ser programados para execução em nós fornecidos por um ou mais grupos de nós de computação. Um cluster AWS PCS pode ter várias filas de trabalhos. Por exemplo, você pode criar uma fila que usa Amazon EC2 On-demand Instances para trabalhos de alta prioridade e outra fila que usa Amazon EC2 Spot Instances para trabalhos de baixa prioridade.

Tópicos

Criando uma fila no AWS PCS

Este tópico fornece uma visão geral das opções disponíveis e descreve o que considerar ao criar uma fila no AWS PCS.

> **Note**
>
> Você pode definir configurações personalizadas do Slurm em filas para implementar políticas de agendamento e gerenciamento de recursos específicos da partição. Para obter mais informações, consulte Definindo configurações personalizadas do Slurm no PCS AWS.

Pré-requisitos

- Um cluster AWS PCS - as filas só podem ser criadas em associação com um cluster AWS PCS específico.
- Um ou mais grupos de nós de computação AWS PCS — uma fila deve estar associada a pelo menos um grupo de nós de computação AWS PCS.

Para criar uma fila no AWS PCS

Você pode criar uma fila usando o Console de gerenciamento da AWS ou o. AWS CLI

Console de gerenciamento da AWS

Para criar uma fila usando o console

1. Abra o console AWS PCS.
2. Selecione o cluster para a fila. Navegue até Filas e escolha Criar fila.
3. Na seção Configuração da fila, forneça os seguintes valores:
   a. Nome da fila — Um nome para sua fila. O nome só pode conter caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um caractere alfabético e não pode ter mais de 25 caracteres. O nome deve ser exclusivo dentro do cluster.
   b. Grupos de nós de computação — Selecione 1 ou mais grupos de nós de computação para atender a essa fila. Um grupo de nós de computação pode ser associado a mais de uma fila.
4. (Opcional) Na seção Configurações adicionais do agendador, você pode adicionar pares de nome e valor do parâmetro para definir configurações adicionais do Slurm. Para obter uma lista completa dos parâmetros compatíveis, consulteConfigurações personalizadas do Slurm para AWS filas PCS.
5. (Opcional) Em Tags, adicione quaisquer tags à sua fila AWS PCS
6. Selecione Criar fila. O campo Status mostrará Criando enquanto o AWS PCS cria a fila. A criação da fila pode levar vários minutos.

Próxima etapa recomendada

- Envie um trabalho para sua nova fila.

AWS CLI

Para criar uma fila usando AWS CLI

Use o comando a seguir para criar sua fila. Faça as seguintes substituições:

1. *region-code*Substitua pela AWS região do cluster. Por exemplo, `us-east-1`

2. *my-queue*Substitua pelo nome da sua fila. O nome só pode conter caracteres alfanuméricos (sensíveis a maiúsculas e minúsculas) e hifens. Ele deve começar com um caractere alfabético e não pode ter mais de 25 caracteres. O nome deve ser exclusivo dentro do cluster.
3. *my-cluster*Substitua pelo nome ou ID do seu cluster.
4. *compute-node-group-id*Substitua pela ID do grupo de nós de computação para atender a fila. Por exemplo, .`pcs_abcdef12345`

> **Note**
>
> Ao criar uma fila, você deve fornecer a ID do grupo de nós de computação e não seu nome.

```
aws pcs create-queue --region region-code \
    --queue-name my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=compute-node-group-id
```

Example— Criação de uma fila com configurações personalizadas do Slurm

```
aws pcs create-queue --region region-code \
    --queue-name my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=compute-node-group-id \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Para obter mais informações, consulte Configurações personalizadas do Slurm para AWS filas PCS.

A criação da fila pode levar alguns minutos. Você pode consultar o status da sua fila com o comando a seguir. Você não poderá enviar trabalhos para a fila até que seu status chegue`ACTIVE`.

```
aws pcs get-queue --region region-code \
    --cluster-identifier my-cluster \
```

```
    --queue-identifier my-queue
```

Próxima etapa recomendada

- Envie um trabalho para sua nova fila

Atualizando uma fila AWS PCS

Este tópico fornece uma visão geral das opções disponíveis e descreve o que considerar ao atualizar uma fila AWS PCS. Para obter informações sobre as configurações personalizadas do Slurm, consulte. Configurações personalizadas do Slurm para AWS filas PCS

Considerações ao atualizar uma fila AWS PCS

As atualizações da fila não afetarão os trabalhos em execução, mas o cluster pode não conseguir aceitar novos trabalhos enquanto a fila estiver sendo atualizada.

Para atualizar uma fila AWS PCS

Você pode usar o Console de gerenciamento da AWS ou AWS CLI para atualizar uma fila.

Console de gerenciamento da AWS

Para atualizar uma fila

1. Abra o console AWS PCS em `https://console.aws.amazon.com/pcs/home#/clusters`
2. Selecione o cluster em que você deseja atualizar uma fila.
3. Navegue até Filas, vá até a fila que deseja atualizar e selecione Editar.
4. Na seção de configuração da fila, atualize qualquer um dos seguintes valores:
    - Grupos de nós — adicione ou remova grupos de nós de computação da associação com a fila.
    - Configurações adicionais do agendador — Adicione, modifique ou remova configurações personalizadas do Slurm para a fila. Para obter mais informações, consulte Configurações personalizadas do Slurm para AWS filas PCS.
    - Tags — Adicione ou remova tags da fila.

5. Selecione Atualizar. O campo Status mostrará Atualizando enquanto as alterações estão sendo aplicadas.

> **Important**
>
> As atualizações da fila podem levar vários minutos.

AWS CLI

Para atualizar uma fila

1. Atualize sua fila com o comando a seguir. Antes da execução do comando, realize as seguintes substituições:

   a. *region-code*Substitua por Região da AWS aquela em que você deseja criar seu cluster.

   b. *my-queue*Substitua pelo nome ou `computeNodeGroupId` pela sua fila.

   c. *my-cluster*Substitua pelo nome ou pelo nome `clusterId` do seu cluster.

   d. Para alterar as associações de grupos de nós de computação, forneça uma lista atualizada para`--compute-node-group-configurations`.

      - Por exemplo, para adicionar um segundo grupo `computeNodeGroupExampleID2` de nós de computação:

```
--compute-node-group-configurations
 computeNodeGroupId=computeNodeGroupExampleID1,computeNodeGroupId=computeNodeGro
```

```
aws pcs update-queue --region region-code \
    --queue-identifier my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=computeNodeGroupExampleID1
```

Example— Atualizar uma fila com configurações personalizadas do Slurm

```
aws pcs update-queue --region region-code \
    --queue-identifier my-queue \
```

```
    --cluster-identifier my-cluster \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Para obter mais informações, consulte Configurações personalizadas do Slurm para AWS filas PCS.

2. A atualização da fila pode levar alguns minutos. Você pode consultar o status da sua fila com o comando a seguir. Você não poderá enviar trabalhos para a fila até que seu status chegue`ACTIVE`.

```
aws pcs get-queue --region region-code \
    --cluster-identifier my-cluster \
    --queue-identifier my-queue
```

Próximas etapas recomendadas

- Envie um trabalho para sua fila atualizada.

Excluindo uma fila no PCS AWS

Este tópico fornece uma visão geral de como excluir uma fila no AWS PCS.

Considerações ao excluir uma fila

- Se houver trabalhos em execução na fila, eles serão encerrados pelo agendador quando a fila for excluída. Os trabalhos pendentes na fila serão cancelados. Considere esperar que os trabalhos na fila sejam concluídos ou manualmente usando stop/cancel os comandos nativos do agendador (como `scancel` para o Slurm).

Excluir a fila

Você pode usar o Console de gerenciamento da AWS ou AWS CLI para excluir uma fila.

Console de gerenciamento da AWS

Para excluir uma fila

1. Abra o console AWS PCS.

2. Selecione o cluster da fila.
3. Navegue até Filas e selecione a fila a ser excluída.
4. Escolha Excluir.
5. O campo Status é exibido`Deleting`. Pode demorar vários minutos para isso ser concluído.

> **Note**
>
> Você pode usar comandos nativos do seu agendador para confirmar que a fila foi excluída. Por exemplo, use `sinfo` ou `squeue` para o Slurm.

AWS CLI

Para excluir uma fila

- Use o comando a seguir para excluir uma fila, com essas substituições:
  - *`region-code`*Substitua por aquele em que Região da AWS seu cluster está.
  - *`my-queue`*Substitua pelo nome ou ID da sua fila.
  - *`my-cluster`*Substitua pelo nome ou ID do seu cluster.

```
aws pcs delete-queue --region region-code \
       --queue-identifier my-queue \
       --cluster-identifier my-cluster
```

Pode levar alguns minutos para excluir a fila.

> **Note**
>
> Você pode usar comandos nativos do seu agendador para confirmar que a fila foi excluída. Por exemplo, use `sinfo` ou `squeue` para o Slurm.

AWS nós de login do PCS

Um cluster AWS PCS geralmente precisa de pelo menos 1 nó de login para oferecer suporte ao acesso interativo e ao gerenciamento de tarefas. Uma forma de fazer isso é com um grupo estático de nós de computação AWS PCS configurado para a capacidade de nó de login. Você também pode configurar uma instância EC2 autônoma para atuar como um nó de login.

Tópicos

Usando um grupo de nós de computação AWS PCS para fornecer nós de login

Este tópico fornece uma visão geral das opções de configuração sugeridas e descreve o que considerar ao usar um grupo de nós de computação do AWS PCS para fornecer acesso persistente e interativo ao seu cluster.

Criação de um grupo de nós de computação AWS PCS para nós de login

Operacionalmente, isso não é muito diferente de criar um grupo normal de nós de computação. No entanto, existem algumas das principais opções de configuração que você pode fazer:

- Defina uma configuração de escalabilidade estática de pelo menos uma instância do EC2 no grupo de nós de computação.
- Escolha a opção de compra sob demanda para evitar que suas instâncias sejam recuperadas.
- Escolha um nome informativo para o grupo de nós de computação, como login.
- Se você quiser que as instâncias do nó de login sejam acessíveis fora da sua VPC, considere usar uma sub-rede pública.
- Se você pretende permitir o acesso SSH, o modelo de lançamento precisará ter um grupo de segurança que exponha a porta SSH aos endereços IP de sua escolha.
- O perfil da instância do IAM deve ter somente as permissões da AWS que você deseja que seus usuários finais tenham. Para mais detalhes, consulte Perfis de instância do IAM para o AWS Parallel Computing Service.

- Considere permitir que o AWS Systems Manager Session Manager gerencie suas instâncias de login.
- Considere restringir o acesso às credenciais da AWS da instância somente para usuários administrativos
- Selecione tipos de instância mais baratos do que para grupos de nós de computação comuns, pois os nós de login serão executados continuamente.
- Use a mesma AMI (ou uma derivada) dos outros grupos de nós de computação para ajudar a garantir que todas as instâncias tenham o mesmo software instalado. Para obter mais informações sobre personalização AMIs, consulte Amazon Machine Images (AMIs) para AWS PCS
- Configure as mesmas montagens do sistema de arquivos de rede (Amazon EFS, Amazon FSx for Lustre etc.) em seus nós de login e em suas instâncias de computação. Para obter mais informações, consulte Usando sistemas de arquivos de rede com AWS PCS.

Acesse seus nós de login

Quando seu novo grupo de nós de computação atingir o status ATIVO, você poderá encontrar as instâncias do EC2 que ele criou e fazer login nelas. Para obter mais informações, consulte Encontrando instâncias de grupos de nós de computação no AWS PCS.

Atualização de um grupo de nós de computação AWS PCS para nós de login

Você pode atualizar um grupo de nós de login usando UpdateComputeNodeGroup o. Como parte do processo de atualização do grupo de nós, as instâncias em execução serão substituídas. Observe que isso interromperá todas as sessões ou processos ativos do usuário na instância. Os trabalhos do Slurm em execução ou em fila não serão afetados. Para obter mais informações, consulte Atualização de um grupo de nós de computação AWS PCS.

Você também pode editar o modelo de execução usado pelo seu grupo de nós de computação. Você deve usar UpdateComputeNodeGroup para aplicar o modelo de execução atualizado ao grupo de nós de computação. As novas instâncias do EC2 lançadas no grupo de nós de computação usam o modelo de execução atualizado. Para obter mais informações, consulte Usando modelos de lançamento do Amazon EC2 com PCS AWS.

Excluindo um grupo de nós de computação AWS PCS para nós de login

Você pode atualizar um grupo de nós de login usando o mecanismo de exclusão de grupos de nós de computação no AWS PCS. As instâncias em execução serão encerradas como parte da exclusão do grupo de nós. Observe que isso interromperá todas as sessões ou processos ativos do usuário na instância. Os trabalhos do Slurm em execução ou em fila não serão afetados. Para obter mais informações, consulte Excluindo um grupo de nós de computação no PCS AWS.

Usando instâncias autônomas como nós de login do AWS PCS

Você pode configurar instâncias EC2 independentes para interagir com o agendador Slurm de um cluster AWS PCS. Isso é útil para criar nós de login, estações de trabalho ou hosts dedicados de gerenciamento de fluxo de trabalho que funcionam com clusters de AWS PCS, mas operam fora do gerenciamento de AWS PCS. Para fazer isso, cada instância autônoma deve:

1. Tenha uma versão compatível do software Slurm instalada.
2. Ser capaz de se conectar ao endpoint Slurmctld do cluster AWS PCS.
3. Configure adequadamente o Slurm Auth e o Cred Kiosk Daemon (`sackd`) com o endpoint e o segredo do cluster PCS. AWS Para obter mais informações, consulte sackd na documentação do Slurm.

Este tutorial ajuda você a configurar uma instância independente que se conecta a um cluster AWS PCS.

Sumário

Etapa 1 — Recupere o endereço e o segredo do cluster AWS PCS de destino

Recupere detalhes sobre o cluster AWS PCS de destino usando AWS CLI o comando a seguir. Antes da execução do comando, realize as seguintes substituições:

- *region-code*Substitua pelo Região da AWS local em que o cluster de destino está sendo executado.
- *cluster-ident*Substitua pelo nome ou identificador do cluster de destino

```
aws pcs get-cluster --region region-code --cluster-identifier cluster-ident
```

O comando retornará uma saída semelhante a este exemplo.

```
{
    "cluster": {
        "name": "get-started",
        "id": "pcs_123456abcd",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
        "status": "ACTIVE",
        "createdAt": "2024-12-17T21:03:52+00:00",
        "modifiedAt": "2024-12-17T21:03:52+00:00",
        "scheduler": {
            "type": "SLURM",
            "version": "25.05"
        },
        "size": "SMALL",
        "slurmConfiguration": {
            "authKey": {
                "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!
slurm-secret-pcs_123456abcd-a12ABC",
                "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
            }
        },
        "networking": {
            "subnetIds": [
                "subnet-0123456789abcdef0"
            ],
            "securityGroupIds": [
                "sg-0123456789abcdef0"
```

```
            ]
        },
        "endpoints": [
            {
                "type": "SLURMCTLD",
                "privateIpAddress": "10.3.149.220",
                "port": "6817"
            }
        ]
    }
}
```

Neste exemplo, o endpoint do controlador Slurm do cluster tem um endereço IP de `10.3.149.220` e está sendo executado na porta. `6817` O `secretArn` será usado em etapas posteriores para recuperar o segredo do cluster. O endereço IP e a porta serão usados em etapas posteriores para configurar o `sackd` serviço.

Etapa 2 — Executar uma instância do EC2

Para iniciar uma instância do EC2

1. Abra o console do Amazon EC2.
2. No painel de navegação, selecione Instances (Instâncias) e, depois, escolha Launch Instances (Iniciar instâncias) para abrir o novo assistente de inicialização de instância.
3. (Opcional) Na seção Nome e tags, forneça um nome para a instância, como`PCS-LoginNode`. O nome é atribuído à instância como uma etiqueta de recurso (`Name=PCS-LoginNode`).
4. Na seção Imagens do aplicativo e do sistema operacional, selecione uma AMI para um dos sistemas operacionais compatíveis com o AWS PCS. Para obter mais informações, consulte Sistemas operacionais compatíveis.
5. Na seção Instance type (Tipo de instância), selecione um tipo de instância compatível. Para obter mais informações, consulte Tipos de instâncias compatíveis.
6. Na seção Par de chaves, selecione o par de chaves SSH a ser usado na instância.
7. Na seção Configurações de rede:

    - Escolha Editar.

        i. Selecione a VPC do seu cluster AWS PCS.

        ii. Em Firewall (grupos de segurança), escolha Selecionar grupo de segurança existente.

A. Selecione um grupo de segurança que permita o tráfego entre a instância e o controlador Slurm do cluster AWS PCS de destino. Para obter mais informações, consulte Requisitos e considerações do grupo de segurança.

B. (Opcional) Selecione um grupo de segurança que permita acesso SSH de entrada à sua instância.

8. Na seção Armazenamento, configure os volumes de armazenamento conforme necessário. Certifique-se de configurar espaço suficiente para instalar aplicativos e bibliotecas para habilitar seu caso de uso.
9. Em Avançado, escolha uma função do IAM que permita acesso ao segredo do cluster. Para obter mais informações, consulte Obtenha o segredo do cluster Slurm.
10. No painel Resumo, escolha Launch instance.

Etapa 3 — Instale o Slurm na instância

Quando a instância for iniciada e ficar ativa, conecte-se a ela usando seu mecanismo preferido. Use o instalador do Slurm fornecido por AWS para instalar o Slurm na instância. Para obter mais informações, consulte Instalador do Slurm.

Baixe o instalador do Slurm, descompacte-o e use o `installer.sh` script para instalar o Slurm. Para obter mais informações, consulte Etapa 3 — Instalar o Slurm.

Etapa 4 — Recuperar e armazenar o segredo do cluster

Essas instruções exigem AWS CLI o. Para obter mais informações, consulte Instalar ou atualizar para a versão mais recente do AWS CLI no Guia AWS Command Line Interface do Usuário da Versão 2.

Armazene o segredo do cluster com os comandos a seguir.

- Crie o diretório de configuração para o Slurm.

```
sudo mkdir -p /etc/slurm
```

- Recupere, decodifique e armazene o segredo do cluster. Antes de executar esse comando, *`region-code`* substitua pela região em que o cluster de destino está sendo executado e *`secret-arn`* substitua pelo valor `secretArn` recuperado na Etapa 1.

```
aws secretsmanager get-secret-value \
 --region region-code \
 --secret-id 'secret-arn' \
 --version-stage AWSCURRENT \
 --query 'SecretString' \
 --output text | base64 -d | sudo tee /etc/slurm/slurm.key
```

> **Warning**
>
> Em um ambiente multiusuário, qualquer usuário com acesso à instância poderá obter o segredo do cluster se puder acessar o serviço de metadados da instância (IMDS). Isso, por sua vez, poderia permitir que eles se passassem por outros usuários. Considere restringir o acesso ao IMDS somente para usuários root ou administrativos. Como alternativa, considere usar um mecanismo diferente que não dependa do perfil da instância para buscar e configurar o segredo.

- Defina a propriedade e as permissões no arquivo de chave do Slurm.

```
sudo chmod 0600 /etc/slurm/slurm.key
sudo chown slurm:slurm /etc/slurm/slurm.key
```

> **Note**
>
> A chave do Slurm deve pertencer ao usuário e ao grupo em que o `sackd` serviço é executado.

Etapa 5 — Configurar a conexão com o cluster AWS PCS

Para estabelecer uma conexão com o cluster AWS PCS, inicie `sackd` como um serviço do sistema seguindo estas etapas.

> **Note**
>
> Se você usa o Slurm 25.05 ou posterior, pode usar um script para configurar seu nó de login para se conectar a vários clusters. Para obter mais informações, consulte Conectando um nó de login independente a vários clusters no AWS PCS.

1. Configure o arquivo de ambiente para o `sackd` serviço com o comando a seguir. Antes de executar o comando, substitua *`ip-address`* e *`port`* pelos valores recuperados dos endpoints na Etapa 1.

```
sudo echo "SACKD_OPTIONS='--conf-server=ip-address:port'" > /etc/sysconfig/sackd
```

2. Crie um arquivo `systemd` de serviço para gerenciar o `sackd` processo.

```
sudo cat << EOF > /etc/systemd/system/sackd.service
[Unit]
Description=Slurm auth and cred kiosk daemon
After=network-online.target remote-fs.target
Wants=network-online.target
ConditionPathExists=/etc/sysconfig/sackd

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/sackd
User=slurm
Group=slurm
RuntimeDirectory=slurm
RuntimeDirectoryMode=0755
ExecStart=/opt/aws/pcs/scheduler/slurm-25.05/sbin/sackd --systemd \$SACKD_OPTIONS
ExecReload=/bin/kill -HUP \$MAINPID
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF
```

3. Defina a propriedade do arquivo `sackd` de serviço.

```
sudo chown root:root /etc/systemd/system/sackd.service && \
    sudo chmod 0644 /etc/systemd/system/sackd.service
```

4. Ative o `sackd` serviço.

```
sudo systemctl daemon-reload && sudo systemctl enable sackd
```

5. Inicie o serviço `sackd`.

```
sudo systemctl start sackd
```

Etapa 6 — (Opcional) Teste a conexão

Confirme se o `sackd` serviço está em execução. Segue um exemplo de saída. Se houver erros, eles geralmente aparecerão aqui.

```
[root@ip-10-3-27-112 ~]# systemctl status sackd
[x] sackd.service - Slurm auth and cred kiosk daemon
   Loaded: loaded (/etc/systemd/system/sackd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-12-17 16:34:55 UTC; 8s ago
 Main PID: 9985 (sackd)
   CGroup: /system.slice/sackd.service
           ##9985 /opt/aws/pcs/scheduler/slurm-25.05/sbin/sackd --systemd --conf-
server=10.3.149.220:6817

Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Starting Slurm auth and cred
 kiosk daemon...
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Started Slurm auth and cred
 kiosk daemon.
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal sackd[9985]: sackd: running
```

Confirme se as conexões com o cluster estão funcionando usando os comandos do cliente Slurm, como e. `sinfo` `squeue` Aqui está um exemplo de saída de`sinfo`.

```
[root@ip-10-3-27-112 ~]# /opt/aws/pcs/scheduler/slurm-25.05/bin/sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
all up infinite 4 idle~ compute-[1-4]
```

Você também deve ser capaz de enviar trabalhos. Por exemplo, um comando semelhante a esse exemplo iniciaria um trabalho interativo em 1 nó no cluster.

```
/opt/aws/pcs/scheduler/slurm-25.05/bin/srun --nodes=1 -p all --pty bash -i
```

Conectando um nó de login independente a vários clusters no AWS PCS

O `pcs-multi-cluster-login-configure.sh` script fornece uma maneira automatizada de configurar vários `sackd` daemons do Slurm em um único nó de login independente. Ele permite que o nó de login se comunique com vários clusters. O script automatiza as seguintes operações:

- Usa ações da API AWS PCS para obter informações do cluster
- Solicita a chave de autenticação Slurm codificada em base64
- Cria um arquivo JWKS do Slurm com chave de autenticação de cluster
- Configura o `sackd` serviço com terminais e portas de cluster
- Cria um arquivo `systemd` de serviço para um daemon específico do cluster `sackd`
- Gera um script de ativação para configuração do ambiente de cluster
- Ativa e inicia o `sackd` serviço

> **Note**
>
> Esse script requer a versão 25.05 ou posterior do Slurm.

O Slurm já deve estar instalado na instância (equivalente à etapa 3 do processo manual). A instância deve ser capaz de alcançar os endpoints do cluster de destino. O script executa as operações equivalentes das etapas 4 e 5 no processo de configuração manual. Ele obtém automaticamente as informações do cluster, configura o `sackd` serviço, cria os arquivos de `systemd` serviço necessários e cria um script de ativação que os usuários podem usar para configurar seu ambiente de shell para interação com o cluster.

Tópicos

Pré-requisitos para o script de configuração do nó de login de vários clusters do AWS PCS

Requisitos do sistema

- Sistema operacional Linux com `systemd` suporte
- Privilégios de root para configuração do sistema

Comandos e pacotes necessários

- `bash`— Interpretador Shell (versão 4.0+)
- `curl`— Para recuperação de AWS metadados do IMDS v2
- `jq`— Processador JSON para analisar respostas AWS da API
- `aws`— AWS CLI v2 para executar ações da API AWS PCS e para acesso ao Secrets Manager
- `systemctl`— gerenciamento `systemd` de serviços
- `find`— Utilitário de pesquisa do sistema de arquivos
- `grep`— Correspondência de padrões de texto
- `sed`— Editor de stream para manipulação de texto
- `sort`— Utilitário de classificação de texto
- `tail`— Exibe as últimas linhas de um arquivo
- `mkdir`— Criação de diretório
- `chmod`— Altera as permissões do arquivo
- `chown`— Altera a propriedade do arquivo
- `ldconfig`— Configuração do vinculador dinâmico

AWS requisitos

- Um cluster AWS PCS que executa o Slurm versão 25.05 ou posterior
- AWS credenciais configuradas (por meio de uma função do IAM, arquivo de credenciais ou variáveis de ambiente)
- Permissões para:
  - `pcs:GetCluster`

- `secretsmanager:GetSecretValue`(se você usar um segredo alternativo)

Usuários e grupos do sistema

- O `slurm` usuário e o grupo devem existir no sistema

Instalação do Slurm

- O Slurm deve ser instalado no mesmo local dos pacotes de instalação do AWS PCS Slurm:

```
/opt/aws/pcs/scheduler/slurm-version
```

AWS Código de script de configuração do nó de login de vários clusters PCS

Salve o código-fonte a seguir em um arquivo com o seguinte nome:

```
pcs-multi-cluster-login-configure.sh
```

Código-fonte do script

```
#!/bin/bash
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# AWS PCS Multi-Cluster Standalone Login Node Configuration Script
#
# This script configures AWS Parallel Computing Service (PCS) multi-cluster stand alone
 login nodes
# by setting up the Slurm authentication and credential kiosk daemon (sackd)
# for connecting to remote PCS clusters.
#
# Prerequisites:
# - AWS CLI configured with appropriate permissions
# - Slurm version 25.05 or later
# - Root privileges for system configuration
# - Network connectivity to AWS PCS endpoints
```

```
set -eo pipefail

# Function to display usage
usage() {
    echo "Usage: $0 --cluster-identifier <cluster-identifier> [--endpoint-url
 <endpoint-url>]"
    echo "       $0 -h|--help"
}

# Function to display help
help() {
    echo "AWS PCS Multi-Cluster Standalone Login Node Configuration Script"
    echo "================================================"
    echo
    echo "This script configures multi-cluster standalone login node for AWS Parallel
 Computing Service (PCS)"
    echo "by setting up the Slurm authentication and credential kiosk daemon (sackd)."
    echo
    usage
    echo
    echo "Options:"
    echo "  --cluster-identifier <id>    AWS PCS cluster identifier (required)"
    echo "  --endpoint-url <url>         Custom PCS endpoint URL (optional)"
    echo "  -h, --help                   Show this help message"
    echo
    echo "Examples:"
    echo "  $0 --cluster-identifier my-pcs-cluster"
    echo
    echo "Note: This script requires root privileges and Slurm version 25.05 or later."
}

# Function to retrieve authentication key
get_auth_key() {
    if [ "$ALTERNATE_SECRET_RETRIEVAL" = "true" ]; then
        echo "Retrieving authentication key from AWS Secrets Manager..." >&2
        local auth_key_arn=$(echo "$CLUSTER_INFO" | jq -r
 '.cluster.slurmConfiguration.authKey.secretArn')
        local auth_key_version=$(echo "$CLUSTER_INFO" | jq -r
 '.cluster.slurmConfiguration.authKey.secretVersion')

        if [ "$auth_key_arn" = "null" ] || [ "$auth_key_version" = "null" ]; then
            echo "Error: Auth key information not found in cluster configuration" >&2
            exit 1
        fi
```

```
        if ! aws secretsmanager get-secret-value --secret-id "$auth_key_arn" --version-
id "$auth_key_version" --query SecretString --output text --region "$REGION" 2>/dev/
null; then
            echo "Error: Failed to retrieve auth key from Secrets Manager" >&2
            exit 1
        fi
    else
        echo "Please enter the base64-encoded Slurm authentication key:" >&2
        echo -n "Base64 of the Slurm secret key: " >&2
        local key
        read -rs key
        echo >&2
        echo "$key"
    fi
}

# Function to get next available SACKD port
get_next_sackd_port() {
    local exclude_file="$1"
    local port=6918
    local used_ports=()

    # Get all currently used SACKD ports into an array
    while IFS= read -r line; do
        used_ports+=("$line")
    done < <(find /etc/sysconfig -name "sackd-pcs-*" ! -path "$exclude_file" \
             -exec grep SACKD_PORT= '{}' ';' 2>/dev/null | \
             sed 's/.*SACKD_PORT=//' | sort -n)

    # Loop through used ports to find first available port
    for used_port in "${used_ports[@]}"; do
        if [ "$port" -lt "$used_port" ]; then
            break
        elif [ "$port" -eq "$used_port" ]; then
            ((port++))
        fi
    done

    echo "$port"
}

# Function to configure cluster
configure_cluster() {
```

```
    mkdir -p /etc/slurm
    SLURM_JWKS_FILE="/etc/slurm/slurm-${CLUSTER_NAME}.jwks"
    echo '{"keys":
[{"alg":"HS256","kty":"oct","kid":"key-'"${CLUSTER_ID}"'","k":"'"${BASE64_SLURM_KEY}"'"}]}'
 | jq -c '.' > "${SLURM_JWKS_FILE}"

    chmod 0600 "$SLURM_JWKS_FILE"
    chown slurm:slurm "$SLURM_JWKS_FILE"

    SLURM_INSTALL_PATH="/opt/aws/pcs/scheduler/slurm-${SLURM_VERSION}"

    SACKD_RUNTIME_DIRECTORY="/run/slurm-${CLUSTER_NAME}"
    mkdir -p "${SACKD_RUNTIME_DIRECTORY}"
    chown slurm:slurm "${SACKD_RUNTIME_DIRECTORY}"

    mkdir -p /etc/sysconfig
    SACKD_SERVICE_NAME="sackd-pcs-${CLUSTER_NAME}"
    SACKD_SERVICE_ENV="/etc/sysconfig/${SACKD_SERVICE_NAME}"
    SACKD_PORT=$(get_next_sackd_port "$SACKD_SERVICE_ENV")
    cat > "${SACKD_SERVICE_ENV}" << EOF
SACKD_OPTIONS='--conf-server=$ENDPOINTS'
SLURM_SACK_JWKS='$SLURM_JWKS_FILE'
RUNTIME_DIRECTORY='$SACKD_RUNTIME_DIRECTORY'
SACKD_PORT=$SACKD_PORT
EOF

    SACKD_SERVICE_PATH="/etc/systemd/system/${SACKD_SERVICE_NAME}.service"

    cat << EOF > "$SACKD_SERVICE_PATH"
[Unit]
Description=Slurm auth and cred kiosk daemon
After=network-online.target remote-fs.target
Wants=network-online.target
ConditionPathExists=${SACKD_SERVICE_ENV}

[Service]
Type=notify
EnvironmentFile=${SACKD_SERVICE_ENV}
User=slurm
Group=slurm
RuntimeDirectory=slurm-${CLUSTER_NAME}
RuntimeDirectoryMode=0755
ExecStart=${SLURM_INSTALL_PATH}/sbin/sackd --systemd \$SACKD_OPTIONS
ExecReload=/bin/kill -HUP \$MAINPID
```

```
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF

    chown root:root "$SACKD_SERVICE_PATH"
    chmod 0644 "$SACKD_SERVICE_PATH"
    systemctl daemon-reload && systemctl enable "$SACKD_SERVICE_NAME"
    systemctl restart "$SACKD_SERVICE_NAME"

    ACTIVATE_SCRIPT="activate-pcs-${CLUSTER_NAME}"
    cat > "$ACTIVATE_SCRIPT" << EOF
# Activate script for Slurm cluster ${CLUSTER_NAME}

# Add Slurm paths
export PATH="${SLURM_INSTALL_PATH}/bin:\$PATH"
export MANPATH="${SLURM_INSTALL_PATH}/share/man:\$MANPATH"
export LD_LIBRARY_PATH="${SLURM_INSTALL_PATH}/lib:\$LD_LIBRARY_PATH"
ldconfig

# Set Slurm configuration
export SLURM_CONF="/run/slurm-${CLUSTER_NAME}/conf/slurm.conf"
export PCS_CLUSTER_NAME="${CLUSTER_NAME}"
export PCS_CLUSTER_IDENTIFIER="${CLUSTER_IDENTIFIER}"
export PCS_CLUSTER_ID="${CLUSTER_ID}"

echo "Activated PCS cluster environment: ${CLUSTER_NAME}"

# Deactivate function
function deactivate-pcs-${CLUSTER_NAME}() {
    export PATH="\$(echo "\$PATH" | sed -e "s|${SLURM_INSTALL_PATH}/bin:||g" -e "s|:
${SLURM_INSTALL_PATH}/bin||g" -e "s|^${SLURM_INSTALL_PATH}/bin\$||")"
    export MANPATH="\$(echo "\$MANPATH" | sed -e "s|${SLURM_INSTALL_PATH}/share/man:||
g" -e "s|:${SLURM_INSTALL_PATH}/share/man||g" -e "s|^${SLURM_INSTALL_PATH}/share/man\
$||")"
    export LD_LIBRARY_PATH="\$(echo "\$LD_LIBRARY_PATH" | sed -e "s|
${SLURM_INSTALL_PATH}/lib:||g" -e "s|:${SLURM_INSTALL_PATH}/lib||g" -e "s|^
${SLURM_INSTALL_PATH}/lib\$||")"
    unset SLURM_CONF
    unset PCS_CLUSTER_NAME
```

```
    unset PCS_CLUSTER_IDENTIFIER
    unset PCS_CLUSTER_ID
    unset -f deactivate-pcs-${CLUSTER_NAME}
    ldconfig
    echo "Deactivated PCS cluster environment: ${CLUSTER_NAME}"
}

export -f deactivate-pcs-${CLUSTER_NAME}

EOF
}

# Main function
main() {
    # Parse arguments
    CLUSTER_IDENTIFIER=""
    PCS_ENDPOINT_URL=""

    while [ "$1" != "" ]; do
        case $1 in
            --cluster-identifier)
                shift
                CLUSTER_IDENTIFIER="$1"
                ;;
            --endpoint-url)
                shift
                PCS_ENDPOINT_URL="--endpoint-url $1"
                ;;
            -h|--help)
                help
                exit 0
                ;;
            *)
                echo "Invalid argument: $1" >&2
                usage >&2
                exit 1
                ;;
        esac
        shift
    done

    # Validate required arguments
    if [ -z "$CLUSTER_IDENTIFIER" ]; then
        echo "Error: --cluster-identifier is required" >&2
```

```
        usage >&2
        exit 1
    fi

    # Validate running as root
    if [ "$EUID" -ne 0 ]; then
        echo "Error: This script must be run as root" >&2
        exit 1
    fi

    # Validate required commands are available
    for cmd in aws jq curl; do
        if ! command -v "$cmd" &> /dev/null; then
            echo "Error: Required command '$cmd' not found" >&2
            exit 1
        fi
    done

    # Get the region name from IMDS v2 with error handling (try IPv6 first, fallback to
 IPv4)
    echo "Retrieving AWS region from instance metadata..."
    # Try IPv6 IMDS endpoint first (fd00:ec2::254) with fast timeout (1s connect, 2s
 total)
    # If IPv6 fails, fallback to IPv4 IMDS endpoint (169.254.169.254)
    IMDS_ENDPOINT="http://[fd00:ec2::254]"
    if ! TOKEN=$(curl -s -X PUT "${IMDS_ENDPOINT}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --connect-timeout 1 --max-time 2 2>/dev/null); then
        IMDS_ENDPOINT="http://169.254.169.254"
        if ! TOKEN=$(curl -s -X PUT "${IMDS_ENDPOINT}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --max-time 5); then
            echo "Error: Failed to retrieve IMDS token. Ensure this script is running
 on an EC2 instance." >&2
            exit 1
        fi
    fi

    if ! REGION=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" "${IMDS_ENDPOINT}/
latest/dynamic/instance-identity/document" --max-time 5 | jq -r '.region'); then
        echo "Error: Failed to retrieve AWS region from instance metadata" >&2
        exit 1
    fi

    echo "Detected AWS region: $REGION"
```

```
   # Retrieve cluster information from AWS PCS
   echo "Retrieving cluster information for: $CLUSTER_IDENTIFIER"
   # shellcheck disable=SC2086
   if ! CLUSTER_INFO=$(aws pcs get-cluster --region "$REGION" --cluster-identifier
"$CLUSTER_IDENTIFIER" $PCS_ENDPOINT_URL 2>/dev/null); then
       echo "Error: Failed to retrieve cluster information. Check cluster identifier
and AWS permissions." >&2
       exit 1
   fi

   CLUSTER_ID=$(echo "$CLUSTER_INFO" | jq -r '.cluster.id')
   CLUSTER_NAME="$(echo "$CLUSTER_INFO" | jq -r '.cluster.name')"
   SLURM_VERSION=$(echo "$CLUSTER_INFO" | jq -r '.cluster.scheduler.version')
   SLURM_VERSION=${SLURM_VERSION#Slurm_}

   # Check if Slurm version is >= 25.05
   # shellcheck disable=SC2072
   if [[ "$SLURM_VERSION" < "25.05" ]]; then
       echo "Error: This script requires Slurm version 25.05 or later. Found version:
$SLURM_VERSION" >&2
       exit 1
   fi

   ENDPOINTS=$(echo "$CLUSTER_INFO" | jq -r '.cluster.endpoints[] | select(.type
== "SLURMCTLD") | (if .privateIpAddress != "" then .privateIpAddress else "["
+ .ipv6Address + "]" end) + ":" + .port' | tr '\n' ',' | sed 's/,$//')

   # Get BASE64_SLURM_KEY
   BASE64_SLURM_KEY=$(get_auth_key)

   if [ -z "$BASE64_SLURM_KEY" ]; then
       echo "Error: base64 Slurm key cannot be empty" >&2
       exit 1
   fi

   configure_cluster

   # Final configuration summary
   echo "========================================"
   echo "Configuration completed successfully!"
   echo "========================================"
   echo "Cluster Name: $CLUSTER_NAME"
   echo "Cluster ID: $CLUSTER_ID"
   echo "Slurm Version: $SLURM_VERSION"
```

```
    echo "Service Name: $SACKD_SERVICE_NAME"
    echo "SACKD Port: $SACKD_PORT"
    echo
    echo "To activate this cluster environment, run:"
    echo "  source ./$ACTIVATE_SCRIPT"
    echo
    echo "To deactivate this cluster environment, run:"
    echo "  deactivate-pcs-${CLUSTER_NAME}"
    echo
    echo "To check service status:"
    echo "  systemctl status $SACKD_SERVICE_NAME"
    echo
    echo "To view service logs:"
    echo "  journalctl -u $SACKD_SERVICE_NAME -f"
}

# Exit if being sourced for testing
[[ "${BASH_SOURCE[0]}" != "${0}" ]] && return

# Execute main function
main "$@"
```

Usando o script de configuração do nó de login de vários clusters do AWS PCS

Executar o script

Para executar o script de configuração

1. Salve o conteúdo do script em um arquivo chamado:

   ```
   pcs-multi-cluster-login-configure.sh
   ```

2. Torne-o executável:

   ```
   chmod +x pcs-multi-cluster-login-configure.sh
   ```

3. Execute o script :

   ```
   ./pcs-multi-cluster-login-configure.sh --cluster-identifier cluster-name
   ```

Ambientes de interação de clusters

Após a configuração bem-sucedida, o script gera um script de ativação específico do cluster no diretório atual. O script tem o nome`activate-pcs-`*`cluster-name`*. O script de ativação configura as variáveis de ambiente e os caminhos necessários para interagir com o cluster de destino.

Para ativar um ambiente de cluster

- Use o `source` comando para executar o script de ativação

  ```
  source ./activate-pcs-cluster-name
  ```

  Example

  ```
  # Activate cluster environment for cluster 'my-cluster'
  source ./activate-pcs-my-cluster

  # Now you can use Slurm commands
  sinfo
  squeue
  sbatch my-job.sh
  ```

O que o script de ativação faz

- Define a variável de `SLURM_CONF` ambiente para apontar para a configuração do cluster.
- Atualiza o `PATH` para incluir os binários do Slurm do cluster.
- Configura outras variáveis de ambiente Slurm necessárias (`MANPATH`,). `LD_LIBRARY_PATH`
- Define as variáveis de identificação do cluster AWS PCS.
- Permite uma interação perfeita com o cluster AWS PCS de destino.

Para desativar um ambiente de cluster

- Execute o comando de desativação.

  ```
  deactivate-pcs-cluster-name
  ```

Example

```
# After activating a cluster
source ./activate-pcs-my-cluster

# Work with the cluster
sinfo

# Deactivate when done
deactivate-pcs-my-cluster
```

O que o comando de desativação faz

- Restaura a variável de PATH ambiente original.
- Desdefine as variáveis de ambiente Slurm específicas do cluster.
- Retorna o ambiente do shell ao estado de pré-ativação.

> **Note**
>
> A ativação é específica da sessão e deve ser originada na sessão do shell em que você deseja interagir com o cluster.

AWS Rede PCS

Seu cluster AWS PCS é criado em uma Amazon VPC. Este capítulo inclui os tópicos a seguir sobre redes para o agendador e os nós do seu cluster.

Com exceção da escolha de uma sub-rede para executar instâncias, você deve usar modelos de EC2 execução para configurar a rede para grupos de nós de computação do AWS PCS. Para obter mais informações sobre modelos de inicialização, consulte Usando modelos de lançamento do Amazon EC2 com PCS AWS.

Tópicos

AWS Requisitos e considerações sobre PCS, VPC e sub-rede

Ao criar um cluster AWS PCS, você especifica uma VPC e uma sub-rede nessa VPC. Este tópico fornece uma visão geral dos requisitos e considerações específicos do AWS PCS para a VPC e a (s) sub-rede (s) que você usa com seu cluster. Se você não tiver uma VPC para usar com o AWS PCS, poderá criar uma usando um modelo fornecido AWS. CloudFormation Para obter mais informações sobre VPCs, consulte Nuvens privadas virtuais (VPC) no Guia do usuário da Amazon VPC.

Requisitos e considerações para VPCs

Quando você cria um cluster, a VPC especificada deve atender aos requisitos e considerações a seguir:

- A VPC deve ter um número suficiente de endereços IP disponíveis para o cluster, todos os nós e outros recursos de cluster que você deseja criar. Para obter mais informações, consulte o endereçamento IP para você VPCs e suas sub-redes no Guia do usuário da Amazon VPC.
- Se o seu cluster usa IPv6:

- Associe um bloco IPv6 CIDR à sua VPC. Para obter mais informações, consulte Criar uma VPC no Guia do usuário da Amazon VPC.

  > **Important**
  >
  > Embora você possa configurar sua VPC com ambos IPv4 e IPv6, você só pode escolher 1 tipo de rede para seu cluster.

- Ative a atribuição automática IPv6 de endereço para suas sub-redes.
- Para obter mais informações, consulte:
  - IPv6 em AWS
  - Entendendo o IPv6 endereçamento na AWS e projetando um plano de endereçamento escalável

- A VPC deve ter um nome de host DNS e suporte à resolução de DNS. Caso contrário, os nós não poderão registrar o cluster do cliente. Para acessar mais informações, consulte Atributos de DNS para sua VPC no Guia do usuário da Amazon VPC.
- A VPC pode exigir o uso de VPC endpoints AWS PrivateLink para poder entrar em contato com a API PCS. AWS Para obter mais informações, consulte Conecte sua VPC aos serviços usados AWS PrivateLink no Guia do usuário da Amazon VPC.

> **Important**
>
> AWS O PCS não oferece suporte a uma VPC com locação de instância dedicada. A VPC que você usa para AWS PCS deve usar a locação de `default` instâncias. Você pode alterar a locação da instância para uma VPC existente. Para obter mais informações, consulte Alterar a locação da instância de uma VPC no Guia do usuário do Amazon Elastic Compute Cloud.

Requisitos e considerações para sub-redes

Quando você cria um cluster Slurm, o AWS PCS cria uma interface de rede elástica (ENI) na sub-rede especificada. Essa interface de rede permite a comunicação entre o controlador do agendador e a VPC do cliente. A interface de rede também permite que o Slurm se comunique com os componentes implantados em sua conta. Você só pode especificar a sub-rede de um cluster no momento da criação.

Requisitos de sub-redes para clusters

A sub-rede que você especifica ao criar um cluster deve atender aos seguintes requisitos:

- A sub-rede deve ter pelo menos 1 endereço IP para ser usada pelo AWS PCS.
- Se seu cluster usa IPv6, todas as sub-redes em seu cluster devem usar. IPv6

> **Important**
>
> Grupos de nós de computação configurados com amostras de AWS PCS AMIs e várias interfaces de rede não funcionarão atualmente se as sub-redes estiverem configuradas apenas para uso. IPv6 Em vez disso, use sub-redes de pilha dupla (IPv4 e IPv6) ou sub-redes somente. IPv4 Para obter mais informações, consulte Usando amostras de Amazon Machine Images (AMIs) com AWS PCS.

- A sub-rede não pode residir em AWS Outposts AWS Wavelength, ou em uma zona AWS local.
- A sub-rede pode ser pública ou privada. Recomendamos que você especifique uma sub-rede privada, se possível. Uma sub-rede pública é uma sub-rede com uma tabela de rotas que inclui uma rota para um gateway da Internet; uma sub-rede privada é uma sub-rede com uma tabela de rotas que não inclui uma rota para um gateway da Internet.

Requisitos de sub-redes para nós

Você pode implantar nós e outros recursos de cluster na sub-rede especificada ao criar seu cluster AWS PCS e em outras sub-redes na mesma VPC.

Qualquer sub-rede na qual você implanta nós e recursos de cluster deve atender aos seguintes requisitos:

- Você deve garantir que a sub-rede tenha endereços IP disponíveis suficientes para implantar todos os nós e recursos do cluster.
- Se seu cluster usa IPv4 e você planeja implantar nós em uma sub-rede pública, essa sub-rede deve atribuir endereços públicos automaticamente IPv4 .

> **Note**
>
> As instâncias em uma sub-rede pública devem usar um grupo de segurança com regras de entrada que permitam o tráfego de endereços IP públicos. A menos que você tenha restrições específicas de endereço de origem, isso significa um endereço de IPv4 origem 0.0.0.0/0 ou um endereço de IPv6 origem de: :/0.

- Se a sub-rede na qual você implanta nós for uma sub-rede privada e sua tabela de rotas não incluir uma rota para um dispositivo de tradução de endereços de rede (NAT) (), adicione endpoints de VPC usando a VPC do cliente. IPv4 AWS PrivateLink Os endpoints VPC são necessários para todos os AWS serviços com os quais os nós entram em contato. O único endpoint necessário é que o AWS PCS permita que o nó chame a ação da `RegisterComputeNodeGroupInstance` API. Para obter mais informações, consulte RegisterComputeNodeGroupInstancea Referência da API AWS PCS.
- O status da sub-rede pública ou privada não afeta o AWS PCS; os endpoints necessários devem estar acessíveis.

Criação de uma VPC para seu AWS cluster PCS

Você pode criar uma Amazon Virtual Private Cloud (Amazon VPC) para seus clusters dentro do AWS Parallel Computing Service (AWS PCS).

Use a Amazon VPC para lançar recursos de VPC em uma rede virtual que você definiu. Essa rede virtual é muito semelhante a uma rede tradicional que pode ser operada no seu próprio data center. Porém, ela vem com os benefícios do uso da infraestrutura escalável da Amazon Web Services. Recomendamos que você tenha uma compreensão completa do serviço Amazon VPC antes de implantar clusters VPC de produção. Para obter mais informações, consulte O que é Amazon VPC? no modo visual do autor. Guia do usuário do Amazon VPC.

Um cluster PCS, nós e recursos de suporte (como sistemas de arquivos e serviços de diretório) são implantados em sua Amazon VPC. Se você quiser usar uma Amazon VPC existente com o PCS, ela deverá atender aos requisitos descritos em. AWS Requisitos e considerações sobre PCS, VPC e sub-rede Este tópico descreve como criar uma VPC que atenda aos requisitos de PCS usando um modelo fornecido AWS. CloudFormation Depois de implantar um modelo, você pode visualizar os recursos criados por ele para saber exatamente quais recursos foram criados e a configuração desses recursos.

Pré-requisitos

Para criar uma Amazon VPC para PCS, você deve ter as permissões do IAM necessárias para criar recursos da Amazon VPC. Esses recursos são sub-redes VPCs, grupos de segurança, tabelas e rotas de rotas e gateways de internet e NAT. Para obter mais informações, consulte Criar uma VPC com uma sub-rede pública no Guia do usuário da Amazon VPC. Para revisar a lista completa do Amazon EC2, consulte Ações, recursos e chaves de condição do Amazon EC2 na Referência de autorização de serviço.

Crie uma Amazon VPC

Crie uma VPC copiando e colando a URL apropriada para Região da AWS onde você usará o PCS. Você também pode baixar o CloudFormation modelo e enviá-lo você mesmo para o CloudFormation console.

- Leste dos EUA (Norte da Virgínia) (us-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Leste dos EUA (Ohio) (us-east-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Oeste dos EUA (Oregon) (us-west-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-west-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Somente modelo

```
https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/
assets/main.yaml
```

Para criar uma Amazon VPC para PCS

1. Abra o modelo no CloudFormation console.

   > **Note**
   >
   > Eles são pré-preenchidos no modelo para que você possa simplesmente deixá-los como valores padrão.

2. Em Forneça um nome de pilha, depois em Nome da pilha, insira. `hpc-networking`
3. Em parâmetros, insira os seguintes detalhes:

   a. Em VPC, em seguida, digite CidrBlock`10.3.0.0/16`

   b. Em Sub-redes A:

      i. Em seguida, CidrPublicSubnetA, insira `10.3.0.0/20`

      ii. Em seguida, CidrPrivateSubnetA, insira `10.3.128.0/20`

   c. Em Sub-redes B:

      i. Em seguida, CidrPublicSubnetB, insira `10.3.16.0/20`

      ii. Em seguida, CidrPrivateSubnetA, insira `10.3.144.0/20`

   d. Em Sub-redes C:

      i. Para ProvisionSubnetsC, selecione`True`.

      > **Note**
      >
      > Se você estiver criando uma VPC em uma região com menos de três zonas de disponibilidade, essa opção será ignorada se definida como. `True`

      ii. Em seguida, CidrPublicSubnetB, insira `10.3.32.0/20`

      iii. Em seguida, CidrPrivateSubnetA, insira `10.3.160.0/20`

4. Em Capacidades, marque a caixa Eu reconheço que a AWS CloudFormation pode criar recursos do IAM.

Monitore o status da CloudFormation pilha. Quando chegar`CREATE_COMPLETE`, o recurso de VPC estará pronto para você usar.

> **Note**
>
> Para ver todos os recursos criados pelo CloudFormation modelo, abra o CloudFormation console. Escolha a pilha `hpc-networking` e depois a guia Resources (Recursos).

Grupos de segurança no AWS PCS

Os grupos de segurança no Amazon EC2 atuam como firewalls virtuais para controlar o tráfego de entrada e saída para as instâncias. Use um modelo de execução para um grupo de nós de computação do AWS PCS para adicionar ou remover grupos de segurança de suas instâncias. Se seu modelo de lançamento não contiver nenhuma interface de rede, use `SecurityGroupIds` para fornecer uma lista de grupos de segurança. Se seu modelo de execução definir interfaces de rede, você deverá usar o `Groups` parâmetro para atribuir grupos de segurança a cada interface de rede. Para obter mais informações sobre modelos de inicialização, consulte Usando modelos de lançamento do Amazon EC2 com PCS AWS.

> **Note**
>
> As alterações na configuração do grupo de segurança no modelo de execução afetam somente as novas instâncias lançadas após a atualização do grupo de nós de computação.

Requisitos e considerações do grupo de segurança

AWS O PCS cria uma interface de rede elástica (ENI) entre contas na sub-rede que você especifica ao criar um cluster. Isso fornece ao agendador de HPC, que está sendo executado em uma conta gerenciada por AWS, um caminho para se comunicar com instâncias do EC2 lançadas pelo PCS. AWS Você deve fornecer um grupo de segurança para essa ENI que permita a comunicação bidirecional entre a ENI do agendador e suas instâncias EC2 de cluster.

Uma maneira simples de fazer isso é criar um grupo de segurança autorreferenciado permissivo que permita o TCP/IP tráfego em todas as portas entre todos os membros do grupo. Você pode anexar isso ao cluster e às instâncias do EC2 do grupo de nós.

Exemplo de configuração permissiva de grupo de segurança

IPv4

Tipo de regra	Protocolos	Portas	Fonte	Destino
Entrada	Todos	Todos	Self	
Saída	Todos	Tudo		0.0.0.0/0
Saída	Todos	Todos		Self

IPv6

Tipo de regra	Protocolos	Portas	Fonte	Destino
Entrada	Todos	Todos	Self	
Saída	Todos	Tudo		::/0
Saída	Todos	Todos		Self

Essas regras permitem que todo o tráfego flua livremente entre o controlador Slurm e os nós, permitem que todo o tráfego de saída chegue a qualquer destino e habilite o tráfego EFA.

Exemplo de configuração restritiva de grupo de segurança

Você também pode limitar as portas abertas entre o cluster e seus nós de computação. Para o agendador do Slurm, o grupo de segurança anexado ao seu cluster deve permitir as seguintes portas:

- 6817 — habilitar conexões de entrada a partir de instâncias `slurmctld` do EC2
- 6818 — habilite conexões de saída `slurmctld` para `slurmd` execução em instâncias do EC2

O grupo de segurança conectado aos seus nós de computação deve permitir as seguintes portas:

- 6817 — habilite conexões de saída a partir de instâncias `slurmctld` do EC2.

- 6818 — habilite conexões de entrada e saída de e para `slurmd` instâncias de `slurmctld` grupos `slurmd` de nós
- 60001—63000 — conexões de entrada e saída entre instâncias de grupos de nós para oferecer suporte `srun`
- Tráfego EFA entre instâncias de grupos de nós. Para obter mais informações, consulte Preparar um grupo de segurança habilitado para EFA no Guia do usuário para instâncias Linux
- Qualquer outro tráfego entre nós exigido pela sua carga de trabalho

Várias interfaces de rede no AWS PCS

Algumas instâncias do EC2 têm várias placas de rede. Isso permite que eles forneçam maior desempenho de rede, incluindo recursos de largura de banda acima de 100 Gbps e melhor manuseio de pacotes. Para obter mais informações sobre instâncias com várias placas de rede, consulte Interfaces de rede elásticas no Guia do usuário do Amazon Elastic Compute Cloud.

Configure placas de rede adicionais para instâncias em um grupo de nós de computação AWS PCS adicionando interfaces de rede ao modelo de execução do EC2. Abaixo está um exemplo de modelo de lançamento que permite duas placas de rede, como as encontradas em uma `hpc7a.96xlarge` instância. Observe os detalhes a seguir:

- A sub-rede de cada interface de rede deve ser a mesma que você escolheu ao configurar o grupo de nós de computação AWS PCS que usará o modelo de execução.
- O dispositivo de rede principal, onde ocorrerá a comunicação de rede de rotina, como tráfego SSH e HTTPS, é estabelecido definindo um `DeviceIndex` de`0`. Outras interfaces de rede têm um `DeviceIndex` de`1`. Só pode haver uma interface de rede primária — todas as outras interfaces são secundárias.
- Todas as interfaces de rede devem ter uma interface exclusiva`NetworkCardIndex`. Uma prática recomendada é numerá-los sequencialmente conforme definidos no modelo de lançamento.
- Os grupos de segurança para cada interface de rede são definidos usando`Groups`. Neste exemplo, um grupo de segurança SSH de entrada (`sg-`*`SshSecurityGroupId`*) é adicionado à interface de rede primária, bem como o grupo de segurança que permite comunicações dentro do cluster (). `sg-`*`ClusterSecurityGroupId`* Finalmente, um grupo de segurança que permite conexões de saída com a Internet (`sg-`*`InternetOutboundSecurityGroupId`*) é adicionado às interfaces primária e secundária.

```
{
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "NetworkCardIndex": 0,
            "SubnetId": "subnet-SubnetId",
            "Groups": [
               "sg-SshSecurityGroupId",
               "sg-ClusterSecurityGroupId",
               "sg-InternetOutboundSecurityGroupId"
            ]
        },
        {
            "DeviceIndex": 1,
            "NetworkCardIndex": 1,
            "SubnetId": "subnet-SubnetId",
            "Groups": ["sg-InternetOutboundSecurityGroupId"]
        }
    ]
}
```

Grupos de posicionamento para instâncias do EC2 no AWS PCS

Você pode usar um grupo de posicionamento para influenciar o posicionamento das instâncias do EC2 de acordo com as necessidades da carga de trabalho executada nelas.

Tipos de grupos de posicionamento

- Cluster — agrupa as instâncias em uma zona de disponibilidade para otimizar a comunicação de baixa latência.
- Partição — distribui instâncias em partições lógicas para ajudar a maximizar a resiliência.
- Spread — impõe rigorosamente que um pequeno número de instâncias seja executado em hardware distinto, o que também pode ajudar na resiliência.

Para obter mais informações, consulte Grupos de posicionamento para suas instâncias do Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.

Recomendamos que você inclua um grupo de posicionamento de cluster ao configurar um grupo de nós de computação AWS PCS para usar o Elastic Fabric Adapter (EFA).

Para criar um grupo de posicionamento de clusters que funcione com o EFA

1. Crie um grupo de posicionamento com o tipo de cluster para o grupo de nós de computação.
   - Use o seguinte AWS CLI comando:

     ```
     aws ec2 create-placement-group --strategy cluster --group-name PLACEMENT-GROUP-NAME
     ```

   - Você também pode usar um CloudFormation modelo para criar um grupo de posicionamento. Para obter mais informações, consulte Como trabalhar com CloudFormation modelos no Guia AWS CloudFormation do usuário. Faça o download do modelo a partir do URL a seguir e faça o upload para o CloudFormation console.

     ```
     https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-placement-group.yaml
     ```

2. Inclua o grupo de posicionamento no modelo de lançamento do EC2 para o grupo de nós de computação AWS PCS.

Usando o Elastic Fabric Adapter (EFA) com PCS AWS

O Elastic Fabric Adapter (EFA) é uma interconexão de rede avançada de alto desempenho AWS que você pode conectar à sua instância do EC2 para acelerar aplicativos de computação de alta performance (HPC) e aprendizado de máquina. Permitir que seus aplicativos sejam executados em um cluster AWS PCS com o EFA envolve a configuração das instâncias do grupo de nós de computação do AWS PCS para usar o EFA da seguinte maneira.

> **Note**
>
> Instale o EFA em uma AMI AWS compatível com PC — A AMI usada no grupo de nós de computação AWS do PCS deve ter o driver EFA instalado e carregado. Para obter informações sobre como criar uma AMI personalizada com o software EFA instalado, consulteImagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

Sumário

- (Opcional) Crie um grupo de colocação
- Crie ou atualize um modelo de lançamento do EC2
- Crie ou atualize grupos de nós de computação para o EFA
- (Opcional) Teste EFA
- (Opcional) Use um CloudFormation modelo para criar um modelo de lançamento habilitado para EFA

Identifique instâncias EC2 habilitadas para EFA

Para usar o EFA, todos os tipos de instância permitidos para um grupo de computação AWS PCS devem oferecer suporte ao EFA e devem ter o mesmo número de v CPUs (e GPUs se apropriado). Para obter uma lista de instâncias habilitadas para EFA, consulte o Elastic Fabric Adapter para cargas de trabalho de HPC e ML no Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud. Você também pode usar o AWS CLI para ver uma lista de tipos de instância compatíveis com o EFA. *region-code*Substitua pelo Região da AWS local em que você usa o AWS PCS, como`us-east-1`.

```
aws ec2 describe-instance-types \
   --region region-code \
   --filters Name=network-info.efa-supported,Values=true \
   --query "InstanceTypes[*].[InstanceType]" \
   --output text | sort
```

> **Note**
>
> Determine quantas interfaces de rede estão disponíveis — Algumas instâncias do EC2 têm várias placas de rede. Isso permite que eles tenham vários EFAs. Para obter mais informações, consulte Várias interfaces de rede no AWS PCS.

Crie um grupo de segurança para apoiar as comunicações da EFA

AWS CLI

Você pode usar o AWS CLI comando a seguir para criar um grupo de segurança que ofereça suporte ao EFA. O comando gera um ID de grupo de segurança. Faça as seguintes substituições:

- *region-code*— Especifique Região da AWS onde você usa o AWS PCS, como`us-east-1`.
- *vpc-id*— Especifique o ID da VPC que você usa para AWS PCS.
- *efa-group-name*— Forneça o nome escolhido para o grupo de segurança.

```
aws ec2 create-security-group \
    --group-name efa-group-name \
    --description "Security group to enable EFA traffic" \
    --vpc-id vpc-id \
    --region region-code
```

Use os comandos a seguir para anexar regras de grupos de segurança de entrada e saída. Faça a seguinte substituição:

- *efa-secgroup-id*— Forneça o ID do grupo de segurança EFA que você acabou de criar.

```
aws ec2 authorize-security-group-ingress \
    --group-id efa-secgroup-id \
    --protocol -1 \
    --source-group efa-secgroup-id

aws ec2 authorize-security-group-egress \
    --group-id efa-secgroup-id \
    --protocol -1 \
    --source-group efa-secgroup-id
```

CloudFormation template

Você pode usar um CloudFormation modelo para criar um grupo de segurança que ofereça suporte ao EFA. Faça o download do modelo a partir do URL a seguir e, em seguida, carregue-o no AWS CloudFormation console.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-sg.yaml
```

Com o modelo aberto no AWS CloudFormation console, insira as seguintes opções.

- Em Forneça um nome de pilha
  - Em Nome da pilha, insira um nome como`efa-sg-stack`.

- Em Parâmetros
  - Em SecurityGroupName, insira um nome como`efa-sg`.
  - Em VPC, selecione a VPC em que você usará o PCS. AWS

Conclua a criação da CloudFormation pilha e monitore seu status. Quando chega ao `CREATE_COMPLETE` EFA, o grupo de segurança está pronto para uso.

(Opcional) Crie um grupo de colocação

Recomendamos que você execute todas as instâncias que usam o EFA em um grupo de posicionamento de cluster para minimizar a distância física entre elas. Crie um grupo de posicionamento para cada grupo de nós de computação em que você planeja usar o EFA. Consulte Grupos de posicionamento para instâncias do EC2 no AWS PCS para criar um grupo de posicionamento para seu grupo de nós de computação.

Crie ou atualize um modelo de lançamento do EC2

As interfaces de rede EFA são configuradas no modelo de lançamento do EC2 para um grupo de nós de computação AWS PCS. Se houver várias placas de rede, várias EFAs podem ser configuradas. O grupo de segurança EFA e o grupo de posicionamento opcional também estão incluídos no modelo de lançamento.

Aqui está um exemplo de modelo de lançamento para instâncias com duas placas de rede, como hpc7a.96xlarge. As instâncias serão lançadas `subnet-`*`SubnetID1`* em um grupo de posicionamento de clusterspg-*`PlacementGroupId1`*.

Grupos de segurança devem ser adicionados especificamente a cada interface EFA. Todo EFA precisa do grupo de segurança que habilita o tráfego do EFA ()`sg-`*`EfaSecGroupId`*. Outros grupos de segurança, especialmente aqueles que lidam com tráfego regular, como SSH ou HTTPS, só precisam estar conectados à interface de rede primária (designada por um `DeviceIndex` de`0`). Os modelos de inicialização em que as interfaces de rede são definidas não oferecem suporte à configuração de grupos de segurança usando o `SecurityGroupIds` parâmetro — você deve definir um valor para `Groups` cada interface de rede configurada.

```
{
    "Placement": {
        "GroupId": "pg-PlacementGroupId1"
    },
```

```
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "InterfaceType": "efa",
            "NetworkCardIndex": 0,
            "SubnetId": "subnet-SubnetId1",
            "Groups": [
                "sg-SecurityGroupId1",
                "sg-EfaSecGroupId"
            ]
        },
        {
            "DeviceIndex": 1,
            "InterfaceType": "efa",
            "NetworkCardIndex": 1,
            "SubnetId": "subnet-SubnetId1"
            "Groups": ["sg-EfaSecGroupId"]
        }
    ]
}
```

Crie ou atualize grupos de nós de computação para o EFA

Seus grupos de nós de computação do AWS PCS devem conter instâncias que tenham o mesmo número de vCPUs, arquitetura de processador e suporte a EFA. Configure o grupo de nós de computação para usar a AMI com o software EFA instalado nela e para usar o modelo de lançamento que configura as interfaces de rede habilitadas para EFA.

(Opcional) Teste EFA

Você pode demonstrar a comunicação habilitada para EFA entre dois nós em um grupo de nós de computação executando o `fi_pingpong` programa, que está incluído na instalação do software EFA. Se esse teste for bem-sucedido, é provável que o EFA esteja configurado corretamente.

Para começar, você precisa de duas instâncias em execução no grupo de nós de computação. Se seu grupo de nós de computação usa capacidade estática, já deve haver instâncias disponíveis. Para um grupo de nós de computação que usa capacidade dinâmica, você pode iniciar dois nós usando o `salloc` comando. Aqui está um exemplo de um cluster com um grupo dinâmico de nós chamado `hpc7g` associado a uma fila chamada`all`.

```
% salloc --nodes 2 -p all
```

```
salloc: Granted job allocation 6
salloc: Waiting for resource configuration
... a few minutes pass ...
salloc: Nodes hpc7g-[1-2] are ready for job
```

Descubra o endereço IP dos dois nós alocados usando`scontrol`. No exemplo a seguir, os endereços são `10.3.140.69` para `hpc7g-1` e `10.3.132.211` para`hpc7g-2`.

```
% scontrol show nodes hpc7g-[1-2]
NodeName=hpc7g-1 Arch=aarch64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=hpc7g
   ActiveFeatures=hpc7g
   Gres=(null)
   NodeAddr=10.3.140.69 NodeHostName=ip-10-3-140-69 Version=25.05.4
   OS=Linux 5.10.218-208.862.amzn2.aarch64 #1 SMP Tue Jun 4 16:52:10 UTC 2024
   RealMemory=124518 AllocMem=0 FreeMem=110763 Sockets=64 Boards=1
   State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=efa
   BootTime=2024-07-02T19:00:09 SlurmdStartTime=2024-07-08T19:33:25
   LastBusyTime=2024-07-08T19:33:25 ResumeAfterTime=None
   CfgTRES=cpu=64,mem=124518M,billing=64
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
   Reason=Maintain Minimum Number Of Instances [root@2024-07-02T18:59:00]
   InstanceId=i-04927897a9ce3c143 InstanceType=hpc7g.16xlarge

NodeName=hpc7g-2 Arch=aarch64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=hpc7g
   ActiveFeatures=hpc7g
   Gres=(null)
   NodeAddr=10.3.132.211 NodeHostName=ip-10-3-132-211 Version=25.05.4
   OS=Linux 5.10.218-208.862.amzn2.aarch64 #1 SMP Tue Jun 4 16:52:10 UTC 2024
   RealMemory=124518 AllocMem=0 FreeMem=110759 Sockets=64 Boards=1
   State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=efa
   BootTime=2024-07-02T19:00:09 SlurmdStartTime=2024-07-08T19:33:25
   LastBusyTime=2024-07-08T19:33:25 ResumeAfterTime=None
   CfgTRES=cpu=64,mem=124518M,billing=64
   AllocTRES=
```

```
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
   Reason=Maintain Minimum Number Of Instances [root@2024-07-02T18:59:00]
   InstanceId=i-0a2c82623cb1393a7 InstanceType=hpc7g.16xlarge
```

Conecte-se a um dos nós (neste caso de exemplo`hpc7g-1`) usando SSH (ou SSM). Observe que esse é um endereço IP interno, portanto, talvez seja necessário se conectar a partir de um dos seus nós de login se usar SSH. Lembre-se também de que a instância precisa ser configurada com uma chave SSH por meio do modelo de execução do grupo de nós de computação.

```
% ssh ec2-user@10.3.140.69
```

Agora, inicie `fi_pingpong` no modo servidor.

```
/opt/amazon/efa/bin/fi_pingpong -p efa
```

Conecte-se à segunda instância (`hpc7g-2`).

```
% ssh ec2-user@10.3.132.211
```

Execute `fi_pingpong` no modo cliente, conectando-se ao servidor ativado`hpc7g-1`. Você deve ver uma saída semelhante ao exemplo abaixo.

```
% /opt/amazon/efa/bin/fi_pingpong -p efa 10.3.140.69

bytes   #sent   #ack     total       time     MB/sec    usec/xfer   Mxfers/sec
64      10      =10      1.2k        0.00s      3.08      20.75       0.05
256     10      =10      5k          0.00s     21.24      12.05       0.08
1k      10      =10      20k         0.00s     82.91      12.35       0.08
4k      10      =10      80k         0.00s    311.48      13.15       0.08
[error] util/pingpong.c:1876: fi_close (-22) fid 0
```

(Opcional) Use um CloudFormation modelo para criar um modelo de lançamento habilitado para EFA

Como há várias dependências na configuração do EFA, foi fornecido um CloudFormation modelo que você pode usar para configurar um grupo de nós de computação. Ele suporta instâncias com

até quatro placas de rede. Para saber mais sobre instâncias com várias placas de rede, consulte Interfaces de rede elásticas no Guia do usuário do Amazon Elastic Compute Cloud.

Faça o download do CloudFormation modelo a partir do URL a seguir e, em seguida, carregue-o no CloudFormation console em Região da AWS que você usa o AWS PCS.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/pcs-lt-efa.yaml
```

Com o modelo aberto no CloudFormation console, insira os valores a seguir. Observe que o modelo fornecerá alguns valores de parâmetros padrão. Você pode deixá-los como valores padrão.

- Em Forneça um nome de pilha
  - Em Nome da pilha, insira um nome descritivo. Recomendamos incorporar o nome que você escolherá para seu grupo de nós de computação do AWS PCS, como. *NODEGROUPNAME*`-efa-lt`
- Em Parâmetros
  - Em NumberOfNetworkCards, escolha o número de placas de rede nas instâncias que estarão em seu grupo de nós.
  - Em VpcId, escolha a VPC em que seu cluster AWS PCS está implantado.
  - Em NodeGroupSubnetId, escolha a sub-rede em seu cluster VPC onde as instâncias habilitadas para EFA serão executadas.
  - Em PlacementGroupName, deixe o campo em branco para criar um novo grupo de posicionamento de cluster para o grupo de nós. Se você tem um grupo de posicionamento existente que deseja usar, insira o nome dele aqui.
  - Em ClusterSecurityGroupId, escolha o grupo de segurança que você está usando para permitir o acesso a outras instâncias no cluster e à API AWS PCS. Muitos clientes escolhem o grupo de segurança padrão em seu cluster VPC.
  - Em SshSecurityGroupId, forneça o ID de um grupo de segurança que você está usando para permitir acesso SSH de entrada aos nós em seu cluster.
  - Para SshKeyName, selecione o par de chaves SSH para acessar os nós em seu cluster.
  - Para LaunchTemplateName, insira um nome descritivo para o modelo de lançamento, como*NODEGROUPNAME*`-efa-lt`. O nome deve ser exclusivo para você Conta da AWS no Região da AWS local em que você usará o AWS PCS.
- Em Capacidades

- Marque a caixa “Eu reconheço que isso AWS CloudFormation pode criar recursos do IAM”.

Monitore o status da CloudFormation pilha. Quando chega, `CREATE_COMPLETE` o modelo de lançamento está pronto para ser usado. Use-o com um grupo de nós de computação AWS PCS, conforme descrito acima emCrie ou atualize grupos de nós de computação para o EFA.

Usando sistemas de arquivos de rede com AWS PCS

Você pode conectar sistemas de arquivos de rede a nós lançados em um grupo de nós de computação do Serviço de Computação AWS Paralela (AWS PCS) para fornecer um local persistente em que dados e arquivos possam ser gravados e acessados. Você pode usar sistemas de arquivos fornecidos por AWS serviços, incluindo Amazon Elastic File System (Amazon EFS), Amazon FSx for Lustre, Amazon FSx for NetApp ONTAP, Amazon FSx for OpenZFS e Amazon File Cache. Você também pode usar sistemas de arquivos autogerenciados, como servidores NFS.

Este tópico aborda considerações e exemplos do uso de sistemas de arquivos de rede com AWS PCS.

Considerações sobre o uso de sistemas de arquivos de rede

Os detalhes da implementação de vários sistemas de arquivos são diferentes, mas há algumas considerações comuns.

- O software do sistema de arquivos relevante deve estar instalado na instância. Por exemplo, para usar o Amazon FSx for Lustre, o Lustre pacote apropriado deve estar presente. Isso pode ser feito incluindo-o no grupo de nós de computação AMI ou usando um script executado na inicialização da instância.
- Deve haver uma rota de rede entre o sistema de arquivos de rede compartilhado e as instâncias do grupo de nós de computação.
- As regras do grupo de segurança para o sistema de arquivos de rede compartilhado e as instâncias do grupo de nós de computação devem permitir conexões com as portas relevantes.
- Você deve manter um namespace consistente de POSIX usuários e grupos em todos os recursos que acessam os sistemas de arquivos. Caso contrário, trabalhos e processos interativos executados em seu cluster PCS poderão encontrar erros de permissão.
- As montagens do sistema de arquivos são feitas usando modelos de EC2 lançamento. Erros ou tempos limite na montagem de um sistema de arquivos de rede podem impedir que as instâncias se tornem disponíveis para executar trabalhos. Isso, por sua vez, pode levar a custos inesperados. Para obter mais informações sobre depuração de modelos de lançamento, consulte. Usando modelos de lançamento do Amazon EC2 com PCS AWS

Exemplo de montagens de rede

Você pode criar sistemas de arquivos usando o Amazon EFS, o Amazon FSx for Lustre, o Amazon FSx for NetApp ONTAP, o Amazon FSx for OpenZFS e o Amazon File Cache. Expanda a seção relevante abaixo para ver um exemplo de cada montagem de rede.

Amazon EFS

Configuração do sistema de arquivos

Criar um sistema de arquivos do Amazon EFS. Certifique-se de que ele tenha um destino de montagem em cada zona de disponibilidade em que você iniciará as instâncias do grupo de nós de computação do PCS. Além disso, certifique-se de que cada destino de montagem esteja associado a um grupo de segurança que permita acesso de entrada e saída das instâncias do grupo de nós de computação do PCS. Para obter mais informações, consulte Montar alvos e grupos de segurança no Guia do usuário do Amazon Elastic File System.

Modelo de execução

Adicione os grupos de segurança da configuração do sistema de arquivos ao modelo de execução que você usará para o grupo de nós de computação.

Inclua dados do usuário que usam o `cloud-config` mecanismo para montar o sistema de arquivos Amazon EFS. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *`mount-point-directory`*— O caminho em cada instância em que você montará o Amazon EFS
- *`filesystem-id`*— O ID do sistema de arquivos do sistema de arquivos EFS

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /mount-point-directory
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
```

```
  - mount -a -t efs defaults

--==MYBOUNDARY==--
```

Amazon FSx para Lustre

Configuração do sistema de arquivos

Crie um sistema de arquivos FSx for Lustre na VPC onde você usará AWS o PCS. Para minimizar as transferências entre zonas, implante em uma sub-rede na mesma zona de disponibilidade em que você iniciará a maioria das instâncias do grupo de nós de computação do PCS. Certifique-se de que o sistema de arquivos esteja associado a um grupo de segurança que permita acesso de entrada e saída das instâncias do grupo de nós de computação do PCS. Para obter mais informações sobre grupos de segurança, consulte Controle de acesso ao sistema de arquivos com o Amazon VPC no Guia do usuário do Amazon FSx for Lustre.

Modelo de execução

Inclua dados do usuário usados `cloud-config` para montar o sistema de arquivos FSx for Lustre. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *`mount-point-directory`*— O caminho em uma instância em que você deseja montar FSx para o Lustre
- *`filesystem-id`*— O ID do sistema de arquivos do sistema de arquivos FSx for Lustre
- *`mount-name`*— O nome da montagem do sistema de arquivos FSx for Lustre
- *`region-code`*— Região da AWS Onde o sistema de arquivos FSx for Lustre é implantado (deve ser o mesmo do seu sistema AWS PCS)
- (Opcional) *`latest`* — Qualquer versão do Lustre compatível com FSx for Lustre

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- amazon-linux-extras install -y lustre=latest
- mkdir -p /mount-point-directory
```

```
- mount -t lustre filesystem-id.fsx.region-code.amazonaws.com@tcp:/mount-name /mount-point-directory

--==MYBOUNDARY==
```

Amazon FSx para NetApp ONTAP

Configuração do sistema de arquivos

Crie um sistema de arquivos Amazon FSx for NetApp ONTAP na VPC onde você usará AWS o PCS. Para minimizar as transferências entre zonas, implante em uma sub-rede na mesma zona de disponibilidade em que você iniciará a maioria das instâncias do grupo de nós de computação do AWS PCS. Certifique-se de que o sistema de arquivos esteja associado a um grupo de segurança que permita acesso de entrada e saída das instâncias do grupo de nós de computação do AWS PCS. Para obter mais informações sobre grupos de segurança, consulte Controle de acesso ao sistema de arquivos com Amazon VPC no Guia do usuário do FSx ONTAP.

Modelo de execução

Inclua dados do usuário usados `cloud-config` para montar o volume raiz de um sistema de arquivos FSx for ONTAP. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *`mount-point-directory`*— O caminho em uma instância em que você deseja montar seu volume FSx for ONTAP
- *`svm-id`*— O ID SVM FSx para o sistema de arquivos ONTAP
- *`filesystem-id`*— O ID do sistema de arquivos do sistema FSx de arquivos ONTAP
- *`region-code`*— Região da AWS Onde o sistema de arquivos FSx for ONTAP está implantado (deve ser o mesmo do seu sistema AWS PCS)
- *`volume-name`*— O nome do volume FSx for ONTAP

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- mkdir -p /mount-point-directory
```

```
- mount -t nfs svm-id.filesystem-id.fsx.region-code.amazonaws.com:/volume-name /mount-point-directory

--==MYBOUNDARY==
```

Amazon FSx para OpenZFS

Configuração do sistema de arquivos

Crie um sistema de arquivos FSx para OpenZFS na VPC onde você usará o PCS. AWS Para minimizar as transferências entre zonas, implante em uma sub-rede na mesma zona de disponibilidade em que você iniciará a maioria das instâncias do grupo de nós de computação do AWS PCS. Certifique-se de que o sistema de arquivos esteja associado a um grupo de segurança que permita acesso de entrada e saída das instâncias do grupo de nós de computação do AWS PCS. Para obter mais informações sobre grupos de segurança, consulte Gerenciando o acesso ao sistema de arquivos com a Amazon VPC no Guia do usuário do FSx OpenZFS.

Modelo de execução

Inclua dados do usuário usados `cloud-config` para montar o volume raiz de um sistema de arquivos FSx para OpenZFS. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *mount-point-directory*— O caminho em uma instância em que você deseja montar seu compartilhamento FSx para OpenZFS
- *filesystem-id*— O ID do sistema de arquivos FSx para o sistema de arquivos OpenZFS
- *region-code*— Região da AWS Onde o sistema de arquivos FSx for OpenZFS está implantado (deve ser o mesmo do seu AWS sistema PCS)

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- mkdir -p /mount-point-directory
- mount -t nfs -o noatime,nfsvers=4.2,sync,rsize=1048576,wsize=1048576 filesystem-id.fsx.region-code.amazonaws.com:/fsx/ /mount-point-directory
```

```
--==MYBOUNDARY==
```

Amazon File Cache

Configuração do sistema de arquivos

Crie um Amazon File Cache na VPC onde você AWS usará o PCS. Para minimizar as transferências entre zonas, escolha uma sub-rede na mesma zona de disponibilidade em que você iniciará a maioria das instâncias do grupo de nós de computação do PCS. Verifique se o cache de arquivos está associado a um grupo de segurança que permite tráfego de entrada e saída na porta 988 entre suas instâncias do PCS e o cache de arquivos. Para obter mais informações sobre grupos de segurança, consulte Controle de acesso ao cache com Amazon VPC no Guia do usuário do Amazon File Cache.

Modelo de execução

Adicione os grupos de segurança da configuração do sistema de arquivos ao modelo de execução que você usará para o grupo de nós de computação.

Inclua dados do usuário usados `cloud-config` para montar o Amazon File Cache. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *`mount-point-directory`*— O caminho em uma instância em que você deseja montar FSx para o Lustre
- *`cache-dns-name`*— O nome do Sistema de Nomes de Domínio (DNS) para o cache de arquivos
- *`mount-name`*— O nome da montagem do cache de arquivos

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- amazon-linux-extras install -y lustre=2.12
- mkdir -p /mount-point-directory
- mount -t lustre -o relatime,flock cache-dns-name@tcp:/mount-name /mount-point-
directory
```

```
--==MYBOUNDARY==
```

Amazon Machine Images (AMIs) para AWS PCS

AWS O PCS trabalha com o AMIs que você fornece, oferecendo grande flexibilidade no software e na configuração encontrados nos nós do seu cluster. Se você estiver testando o AWS PCS, poderá usar uma amostra de AMI fornecida e mantida pela AWS. Se você estiver usando o AWS PCS na produção, recomendamos que você crie o seu próprio AMIs. Este tópico aborda como descobrir e usar a amostra AMIs, bem como criar e usar sua própria amostra personalizada AMIs.

Tópicos

Usando amostras de Amazon Machine Images (AMIs) com AWS PCS

A AWS fornece uma amostra AMIs que você pode usar como ponto de partida para trabalhar com o AWS PCS.

> **Important**
>
> AMIs As amostras são para fins de demonstração e não são recomendadas para cargas de trabalho de produção.

> **Important**
>
> Grupos de nós de computação configurados com amostras de AWS PCS AMIs e várias interfaces de rede não funcionarão atualmente se as sub-redes estiverem configuradas apenas para uso. IPv6 Em vez disso, use sub-redes de pilha dupla (IPv4 e IPv6) ou sub-redes somente. IPv4

Encontre a amostra atual do AWS PCS AMIs

Console de gerenciamento da AWS

A amostra do AWS PCS AMIs tem a seguinte convenção de nomenclatura:

```
aws-pcs-sample_ami-OS-architecture-scheduler-scheduler-major-version
```

Valores aceitos

- *OS* – amzn2
- *architecture*: x86_64 ou arm64
- *scheduler* – slurm
- *scheduler-major-version* – 25.05

Para encontrar uma amostra de AWS PCS AMIs

1. Abra o EC2 console da Amazon.
2. Acesse AMIs.
3. Escolha Imagens públicas.
4. Em Localizar AMI por atributo ou tag, pesquise uma AMI usando o nome do modelo.

   Exemplos

   - Exemplo de AMI para Slurm 25.05 em instâncias Arm64

     ```
     aws-pcs-sample_ami-amzn2-arm64-slurm-25.05
     ```

   - Exemplo de AMI para Slurm 25.05 em instâncias x86

     ```
     aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05
     ```

   > **Note**
   >
   > Se houver várias AMIs, use a AMI com o carimbo de data/hora mais recente.

5. Use o ID da AMI ao criar ou atualizar um grupo de nós de computação.

AWS CLI

Você pode encontrar o exemplo de AMI de AWS PCS mais recente com os comandos a seguir. *region-code*Substitua pelo Região da AWS local em que você usa o AWS PCS, como`us-east-1`.

- x86_64

```
aws ec2 describe-images --region region-code --owners amazon \
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05*' \
            'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

- Arm 64

```
aws ec2 describe-images --region region-code --owners amazon \
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-arm64-slurm-25.05*' \
            'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

Use o ID da AMI ao criar ou atualizar um grupo de nós de computação.

Saiba mais sobre a amostra AWS PCS AMIs

Para ver o conteúdo e os detalhes de configuração das versões atuais e anteriores da amostra AWS PCS AMIs, consulteNotas de lançamento da amostra AWS PCS AMIs.

Crie seu próprio AMIs compatível com AWS PCS

Para saber como criar seus próprios AMIs que funcionem com o AWS PCS, consulteImagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS

AWS O PCS foi projetado para funcionar com Amazon Machine Images (AMI) que você traz para o serviço. Eles AMIs podem ter software e configurações arbitrários instalados neles, desde

que tenham o agente AWS PCS e uma versão compatível do Slurm instalados e configurados corretamente. Você deve usar os AWS instaladores fornecidos para instalar o software AWS PCS em sua AMI personalizada. Recomendamos que você use AWS instaladores fornecidos para instalar o Slurm em sua AMI personalizada, mas você pode instalar o Slurm sozinho se preferir (não recomendado).

> **Note**
>
> Se quiser experimentar o AWS PCS sem criar uma AMI personalizada, você pode usar uma amostra de AMI fornecida pela AWS. Para obter mais informações, consulte Usando amostras de Amazon Machine Images (AMIs) com AWS PCS.

> **Important**
>
> AWS Atualmente, o PCS requer um kernel com IPv4 suporte para comunicação com nós locais, mesmo quando você usa o AWS PCS em uma rede IPv6 somente.

Este tutorial ajuda você a criar uma AMI que pode ser usada com grupos de nós de computação do PCS para potencializar sua HPC e AI/ML suas cargas de trabalho.

Tópicos

Etapa 1 — Executar uma instância temporária

Execute uma instância temporária que você possa usar para instalar e configurar o software AWS PCS e o agendador Slurm. Você usa essa instância para criar uma AMI compatível com AWS PCS.

Para executar uma instância temporária

1. Abra o console do Amazon EC2.
2. No painel de navegação, escolha Instâncias e, em seguida, escolha Launch instances para abrir o novo assistente de instância de inicialização.
3. (Opcional) Na seção Nome e tags, forneça um nome para a instância, como`PCS-AMI-instance`. O nome é atribuído à instância como uma etiqueta de recurso (`Name=PCS-AMI-instance`).
4. Na seção Application and OS Images (Imagens de aplicação e sistema operacional), selecione uma AMI para um dos sistemas operacionais compatíveis.
5. Na seção Instance type (Tipo de instância), selecione um tipo de instância compatível.
6. Na seção Key pair (Par de chaves), selecione o par de chaves a ser usado na instância.
7. Na seção Configurações de rede:
   - Para Firewall (grupos de segurança), escolha Selecionar grupo de segurança existente e, em seguida, selecione um grupo de segurança que permita acesso SSH de entrada à sua instância.
8. Na seção Storage (Armazenamento), configure os volumes conforme necessário. Certifique-se de configurar espaço suficiente para instalar seus próprios aplicativos e bibliotecas.
9. No painel Resumo painel, escolha Iniciar instância.

Etapa 2 — Instalar o agente AWS PCS

Instale o agente que configura as instâncias iniciadas pelo AWS PCS para uso com o Slurm. Para obter mais informações sobre o agente AWS PCS, consulteAWS Versões do agente PCS.

Para instalar o agente AWS PCS

1. Conecte à instância que você iniciou. Para obter mais informações, consulte Conectar-se à instância do Linux.
2. (Opcional) Para garantir que todos os seus pacotes de software estejam atualizados, faça uma rápida atualização de software na sua instância. esse processo pode demorar alguns minutos.
   - Amazon Linux 2, Amazon Linux 2023, RHEL 9, RHEL 8, Rocky Linux 9 e Rocky Linux 8

```
sudo yum update -y
```

- Ubuntu 22.04 e Ubuntu 24.04

```
sudo apt-get update && sudo apt-get upgrade -y
```

3. Reinicialize a instância e reconecte-se a ela.
4. Baixe os arquivos de instalação do agente AWS PCS. Os arquivos de instalação são empacotados em um arquivo tarball () `.tar.gz` compactado. Para fazer download da última versão estável, use o seguinte comando: *`region`*Substitua pelo Região da AWS local em que você iniciou sua instância temporária, como`us-east-1`.

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-
agent-v1.3.2-1.tar.gz -o aws-pcs-agent-v1.3.2-1.tar.gz
```

Você também pode obter a versão mais recente substituindo o número da versão pelo comando anterior (por exemplo:`aws-pcs-agent-v1-latest.tar.gz`). `latest`

> **Note**
>
> Isso pode mudar em futuras versões do software do agente AWS PCS.

5. (Opcional) Verifique a autenticidade e a integridade do pacote de software AWS PCS. Recomendamos que você faça isso para verificar a identidade do fornecedor do software e para verificar se a aplicação não foi alterada ou corrompida desde que foi publicada.

   a. Baixe a chave GPG pública para AWS PCS e importe-a para o seu chaveiro. *`region`*Substitua pelo Região da AWS local em que você iniciou sua instância temporária. O comando deve retornar um valor de chave. Registre o valor da chave; você o usa na próxima etapa.

   ```
   wget https://aws-pcs-repo-public-keys-region.s3.region.amazonaws.com/aws-pcs-
   public-key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. Execute o comando a seguir para verificar a impressão digital da chave GPG.

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   O comando deve retornar uma impressão digital idêntica à seguinte:

```
1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
```

> **Important**
>
> Não execute o script de instalação do agente AWS PCS se a impressão digital não corresponder. Entrar em contato com o AWS Support.

c. Baixe o arquivo de assinatura e verifique a assinatura do arquivo tarball do software AWS PCS. *region*Substitua pelo Região da AWS local em que você iniciou sua instância temporária, como`us-east-1`.

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-agent-v1.3.2-1.tar.gz.sig
```

A saída deve ser semelhante ao seguinte:

```
gpg: assuming signed data in './aws-pcs-agent-v1.3.2-1.tar.gz'
gpg: Signature made Thu 06 Nov 2025 11:10:36 AM CET using RSA key ID ECC0AE5C
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   Subkey fingerprint: B7E1 8788 3517 6A74 C3D5  EAF5 6088 136D ECC0 AE5C
```

Se o resultado incluir `Good signature` e a impressão digital corresponder à impressão digital retornada na etapa anterior, vá para a próxima etapa.

> **Important**
>
> Não execute o script de instalação do software AWS PCS se a impressão digital não corresponder. Entrar em contato com o AWS Support.

6. Extraia os arquivos do `.tar.gz` arquivo compactado e navegue até o diretório extraído.

```
tar -xf aws-pcs-agent-v1.3.2-1.tar.gz && \
    cd aws-pcs-agent
```

7. Instale o software AWS PCS.

```
sudo ./installer.sh
```

8. Verifique o arquivo da versão do software AWS PCS para confirmar uma instalação bem-sucedida.

```
cat /opt/aws/pcs/version
```

A saída deve ser semelhante ao seguinte:

```
AGENT_INSTALL_DATE='Fri Dec 13 12:28:43 UTC 2024'
AGENT_VERSION='1.3.2'
AGENT_RELEASE='1'
```

Etapa 3 — Instalar o Slurm

Instale uma versão do Slurm compatível com AWS o PCS. Para obter mais informações, consulte Versões Slurm no PCS AWS.

> **Note**
>
> Se você tiver uma AMI com uma versão anterior do software Slurm instalada, deverá executar as etapas a seguir para instalar a nova versão do Slurm. O agente AWS PCS habilita a versão correta dos binários do Slurm em tempo de execução, de acordo com a versão do Slurm configurada no momento da criação do cluster.

Para instalar o Slurm

1. Conecte-se à mesma instância temporária em que você instalou o software AWS PCS.
2. Baixe o software instalador do Slurm. O instalador do Slurm é empacotado em um arquivo tarball () compactado. `.tar.gz` Para fazer download da última versão estável, use o seguinte comando: *`region`*Substitua pela Região da AWS da sua instância temporária, como`us-east-1`.

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz \
```

```
    -o aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz
```

Você também pode obter a versão mais recente substituindo o número da versão pelo comando anterior (por exemplo:`aws-pcs-slurm-25.05-installer-latest.tar.gz`). `latest` Para obter uma lista completa das versões disponíveis com somas de verificação, consulte. Versões Slurm no PCS AWS

> **Note**
>
> Isso pode mudar em futuras versões do software instalador Slurm.

3. (Opcional) Verifique a autenticidade e a integridade do pacote de instalação do Slurm. Recomendamos que você faça isso para verificar a identidade do fornecedor do software e para verificar se a aplicação não foi alterada ou corrompida desde que foi publicada.

   a. Baixe a chave GPG pública para AWS PCS e importe-a para o seu chaveiro. *`region`*Substitua pelo Região da AWS local em que você iniciou sua instância temporária. O comando deve retornar um valor de chave. Registre o valor da chave; você o usa na próxima etapa.

   ```
   wget https://aws-pcs-repo-public-keys-region.s3.region.amazonaws.com/aws-pcs-public-key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. Execute o comando a seguir para verificar a impressão digital da chave GPG.

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   O comando deve retornar uma impressão digital idêntica à seguinte:

   ```
   1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   ```

   > **Important**
   >
   > Não execute o script de instalação do Slurm se a impressão digital não corresponder. Entrar em contato com o AWS Support.

c. Baixe o arquivo de assinatura e verifique a assinatura do arquivo tarball do instalador do Slurm. *region*Substitua pelo Região da AWS local em que você iniciou sua instância temporária, como`us-east-1`.

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz.sig && \
     gpg --verify ./aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz.sig
```

A saída deve ser semelhante ao seguinte:

```
gpg: assuming signed data in './aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz'
gpg: Signature made Fri 24 Oct 2025 05:05:11 PM UTC using RSA key ID ECC0AE5C
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
     Subkey fingerprint: B7E1 8788 3517 6A74 C3D5  EAF5 6088 136D ECC0 AE5C
```

Se o resultado incluir `Good signature` e a impressão digital corresponder à impressão digital retornada na etapa anterior, vá para a próxima etapa.

> **Important**
>
> Não execute o script de instalação do Slurm se a impressão digital não corresponder. Entrar em contato com o AWS Support.

4. Extraia os arquivos do arquivo compactado `.tar.gz` e navegue para o diretório extraído.

```
tar -xf aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz && \
    cd aws-pcs-slurm-25.05-installer
```

5. Instale o Slurm. O instalador baixa, compila e instala o Slurm e suas dependências. Isso leva vários minutos, dependendo das especificações da instância temporária que você selecionou.

```
sudo ./installer.sh -y
```

6. Verifique o arquivo da versão do agendador para confirmar a instalação.

```
cat /opt/aws/pcs/scheduler/slurm-25.05/version
```

A saída deve ser semelhante ao seguinte:

```
SLURM_INSTALL_DATE='Mon Nov 3 14:23:38 UTC 2025'
SLURM_VERSION='25.05.4'
PCS_SLURM_RELEASE='1'
```

Etapa 4 — (Opcional) Instale drivers, bibliotecas e software aplicativo adicionais

Instale drivers, bibliotecas e aplicativos adicionais na instância temporária. Os procedimentos de instalação variam de acordo com os aplicativos e bibliotecas específicos. Se você ainda não criou uma AMI personalizada para AWS PCS, recomendamos que primeiro crie e teste uma AMI com apenas o software AWS PCS e o Slurm instalados e, em seguida, adicione incrementalmente seu próprio software e configurações depois de confirmar o sucesso inicial.

Exemplos

- Software Elastic Fabric Adapter (EFA). Para obter mais informações, consulte Comece a usar o EFA e o MPI para cargas de trabalho de HPC no Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.
- Cliente Amazon Elastic File System (Amazon EFS). Para obter mais informações, consulte Instalação manual do cliente Amazon EFS no Guia do usuário do Amazon Elastic File System.
- Cliente Lustre, para usar o Amazon FSx for Lustre e o Amazon File Cache. Para obter mais informações, consulte Instalando o cliente Lustre no Guia do FSx usuário do Lustre.
- CloudWatch Agente da Amazon, para usar CloudWatch registros e métricas. Para obter mais informações, consulte Instalar o CloudWatch agente no Guia CloudWatch do usuário da Amazon.
- AWS Neuron, para usar os tipos de instância trn* e inf*. Para obter mais informações, consulte a documentação do AWS Neuron.
- Driver NVIDIA, CUDA e DCGM, para usar os tipos de instância p* ou g*.

Etapa 5 — Crie uma AMI compatível com AWS PCS

Depois de instalar os componentes de software necessários, você cria uma AMI que pode ser reutilizada para iniciar instâncias em grupos de nós de computação do AWS PCS.

> **Important**
>
> AWS Atualmente, o PCS requer um kernel com IPv4 suporte para comunicação com nós locais, mesmo quando você usa o AWS PCS em uma rede IPv6 somente.

Para criar uma AMI a partir de sua instância temporária

1. Abra o console do Amazon EC2.
2. No painel de navegação, escolha Instâncias.
3. Selecione a instância temporária que você criou. Escolha Ações, Imagem, Criar imagem.
4. Em Create image (Criar imagem), faça o seguinte:
    a. Em Image name (Nome da imagem), insira um nome descritivo para a AMI.
    b. (Opcional) Em Image description (Descrição da imagem), informe a descrição do propósito da AMI.
    c. Escolha Create Image (Criar imagem).
5. No painel de navegação, escolha AMIs.
6. Localize a AMI que você criou na lista. Aguarde até que seu status mude de Pendente para Disponível e use-o com um grupo de nós de computação AWS PCS.

Etapa 6 — Use a AMI personalizada com um grupo de nós de computação AWS PCS

Você pode usar sua AMI personalizada com um grupo de nós de computação AWS PCS novo ou existente.

> **Important**
>
> AWS Atualmente, o PCS requer um kernel com IPv4 suporte para comunicação com nós locais, mesmo quando você usa o AWS PCS em uma rede IPv6 somente.

New compute node group

Para usar a AMI personalizada

1. Abra o console AWS PCS.
2. No painel de navegação, escolha Clusters.
3. Escolha o cluster em que você usará a AMI personalizada e selecione grupos de nós de computação.
4. Crie um novo grupo de nós de computação. Para obter mais informações, consulte Criação de um grupo de nós de computação no AWS PCS. Em ID da AMI, pesquise o nome ou ID da AMI personalizada que você deseja usar. Conclua a configuração do grupo de nós de computação e escolha Criar grupo de nós de computação.
5. (Opcional) Confirme se a AMI oferece suporte a lançamentos de instâncias. Execute uma instância no grupo de nós de computação. Você pode fazer isso configurando o grupo de nós de computação para ter uma única instância estática ou enviar um trabalho para uma fila que usa o grupo de nós de computação.

    a. Verifique o console do Amazon EC2 até que uma instância apareça marcada com o novo ID do grupo de nós de computação. Para obter mais informações sobre isso, consulteEncontrando instâncias de grupos de nós de computação no AWS PCS..

    b. Ao ver uma instância ser iniciada e concluir o processo de bootstrap, confirme se ela está usando a AMI esperada. Para fazer isso, selecione a instância e, em seguida, inspecione a ID da AMI em Detalhes. Ela deve corresponder à AMI que você configurou nas configurações do grupo de nós de computação.

    c. (Opcional) Atualize a configuração de escalabilidade do grupo de nós de computação de acordo com seus valores preferidos.

Existing compute node group

Para usar a AMI personalizada

1. Abra o console AWS PCS.
2. No painel de navegação, escolha Clusters.
3. Escolha o cluster em que você usará a AMI personalizada e selecione grupos de nós de computação.

4. Selecione o grupo de nós que você deseja configurar e escolha Editar. Em ID da AMI, pesquise o nome ou ID da AMI personalizada que você deseja usar. Conclua a configuração do grupo de nós de computação e escolha Atualizar. As novas instâncias lançadas no grupo de nós de computação usarão a ID da AMI atualizada. As instâncias existentes continuarão usando a AMI antiga até que o AWS PCS as substitua. Para obter mais informações, consulte Atualização de um grupo de nós de computação AWS PCS.
5. (Opcional) Confirme se a AMI oferece suporte a lançamentos de instâncias. Execute uma instância no grupo de nós de computação. Você pode fazer isso configurando o grupo de nós de computação para ter uma única instância estática ou enviar um trabalho para uma fila que usa o grupo de nós de computação.

    a. Verifique o console do Amazon EC2 até que uma instância apareça marcada com o novo ID do grupo de nós de computação. Para obter mais informações sobre isso, consulteEncontrando instâncias de grupos de nós de computação no AWS PCS..

    b. Ao ver uma instância ser iniciada e concluir o processo de bootstrap, confirme se ela está usando a AMI esperada. Para fazer isso, selecione a instância e, em seguida, inspecione a ID da AMI em Detalhes. Ela deve corresponder à AMI que você configurou nas configurações do grupo de nós de computação.

    c. (Opcional) Atualize a configuração de escalabilidade do grupo de nós de computação de acordo com seus valores preferidos.

Etapa 7 — Encerrar a instância temporária

Depois de confirmar que sua AMI funciona conforme o esperado com o AWS PCS, você pode encerrar a instância temporária para parar de incorrer em cobranças por ela.

Para encerrar a instância temporária

1. Abra o console do Amazon EC2.
2. No painel de navegação, escolha Instâncias.
3. Selecione a instância temporária que você criou e escolha Ações, Estado da instância, Encerrar instância.
4. Quando solicitado a confirmar, escolha Encerrar.

Instaladores de software para criar de forma personalizada AMIs para AWS PCS

AWS fornece um arquivo para download que pode instalar o software AWS PCS em uma instância. AWS também fornece software que pode baixar, compilar e instalar versões relevantes do Slurm e de suas dependências. Você pode usar essas instruções para criar uma versão personalizada AMIs para uso com o AWS PCS ou pode usar seus próprios métodos.

Sumário

AWS Instalador do software do agente PCS

O instalador do software do agente AWS PCS configura uma instância para funcionar com o AWS PCS durante o processo de inicialização da instância. Você deve usar AWS instaladores fornecidos para instalar o agente AWS PCS em sua AMI personalizada.

Para obter mais informações sobre o software do agente AWS PCS, consulteAWS Versões do agente PCS.

Instalador do Slurm

O instalador do Slurm baixa, compila e instala versões relevantes do Slurm e de suas dependências. Você pode usar o instalador do Slurm para criar de forma personalizada AMIs para AWS PCS. Você também pode usar seus próprios mecanismos se eles forem consistentes com a configuração de software fornecida pelo instalador do Slurm. Para obter mais informações sobre o suporte do AWS PCS para o Slurm, consulte. Versões Slurm no PCS AWS

O software AWS fornecido instala o seguinte:

- Slurm na versão principal e de manutenção solicitada (atualmente versão 25.05.x) - Licença GPL 2
  - O Slurm é construído com `--sysconfdir` um conjunto de `/etc/slurm`

  - O Slurm é construído com a opção e `--enable-pam --without-munge`
  - O Slurm é construído com a opção `--sharedstatedir=/run/slurm/`
  - O Slurm é construído com suporte a PMIX e JWT
  - O Slurm é instalado em `/opt/aws/pcs/schedulers/slurm-25.05`
- OpenPmix (versão 4.2.6) — Licença
  - O OpenPmix é instalado como um subdiretório do `/opt/aws/pcs/scheduler/`
- libjwt (versão 1.17.0) — Licença MPL-2.0
  - libjwt é instalado como um subdiretório do `/opt/aws/pcs/scheduler/`

O software AWS fornecido altera a configuração do sistema da seguinte forma:

- O `systemd` arquivo Slurm criado pela compilação é copiado `/etc/systemd/system/` com o nome do arquivo. `slurmd-25.05.service`
- Se eles não existirem, um usuário e um grupo (`slurm:slurm`) do Slurm são criados com UID/GID of. `401`
- A pasta `/etc/aws/pcs/scheduler/slurm-25.05/plugstack.conf.d/` é criada para armazenar sua Estenda a funcionalidade do Slurm no AWS PCS com plug-ins SPANK configuração.
- No Amazon Linux 2 e no Rocky Linux 9, a instalação adiciona o repositório EPEL para instalar o software necessário para criar o Slurm ou suas dependências.
- RHEL9 Na instalação, habilitará `codeready-builder-for-rhel-9-rhui-rpms` e `epel-release-latest-9` instalará o software necessário `fedoraproject` para criar o Slurm ou suas dependências.

Sistemas operacionais compatíveis

Consulte Sistemas operacionais compatíveis no AWS PCS.

> **Note**
>
> AMIs de deep learning da AWS As versões (DLAMI) baseadas no Amazon Linux 2 e no Ubuntu 22.04 devem ser compatíveis com o software PCS e os instaladores AWS do Slurm. Para obter mais informações, consulte Como escolher sua DLAMI no AMIs de deep learning da AWS Guia do desenvolvedor.

Tipos de instâncias compatíveis

AWS O software PCS e os instaladores do Slurm oferecem suporte a qualquer tipo de instância x86_64 ou arm64 que possa executar um dos sistemas operacionais compatíveis.

Versões do Slurm suportadas

Consulte Versões Slurm no PCS AWS.

Verifique os instaladores usando uma soma de verificação

Você pode usar SHA256 somas de verificação para verificar os arquivos tarball (.tar.gz) do instalador. Recomendamos que você faça isso para verificar a identidade do fornecedor do software e para verificar se a aplicação não foi alterada ou corrompida desde que foi publicada.

Para verificar um tarball

Use o utilitário sha256sum para a soma de SHA256 verificação e especifique o nome do arquivo tarball. Você deve executar o comando a partir do diretório em que salvou o arquivo tarball.

- SHA256

```
$ sha256sum tarball_filename.tar.gz
```

O comando deve retornar um valor de soma de verificação no formato a seguir.

```
checksum_value tarball_filename.tar.gz
```

Compare o valor da soma de verificação retornado pelo comando com o valor da soma de verificação fornecido na tabela a seguir. Se as somas de verificação corresponderem, é seguro executar o script de instalação.

> **Important**
>
> Se as somas de verificação não corresponderem, não execute o script de instalação. Entre em contato com a Suporte.

Por exemplo, o comando a seguir gera a SHA256 soma de verificação para o tarball do Slurm 25.05.4-1.

```
$ sha256sum aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz
```

Resultado do exemplo:

```
3b0f93bce441d4f4f6935175f2c1e81cd961cb923adb416fa6689f5592047a7d aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz
```

As tabelas a seguir listam as somas de verificação das versões recentes dos instaladores. *us-east-1*Substitua pelo Região da AWS local em que você usa o AWS PCS.

AWS Agente PCS

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
AWS Agente PCS 1.3.2-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz`	`06b32a952a1c849e3442e35c28ac2e4d6962b09286cad748f3c83d561b52ec6f`
AWS Agente PCS 1.3.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.1-1.tar.gz`	`5b7f1eb7b3a86bd2d331b5cb0138d868dc9452da34b480becd86af892c7e8d19`
AWS Agente PCS 1.3.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.0-1.tar.gz`	`eadc9b65c3db248bdde2a6c41814dfb1b97239f24ad55e03d8526dd9ab4a8d16`

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
AWS Agente PCS 1.2.2-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.2-1.tar.gz`	`fd7b6ea5442db75d723fc4971781ce6ae511baa21b87c4286fc1df8127b282b8`
AWS Agente PCS 1.2.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.1-1.tar.gz`	`2b784643ca01ccca1baa64fbfb34bb41efe8bdca69470998b74ce3962bc271d4`
AWS Agente PCS 1.2.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.0-1.tar.gz`	`470db8c4fc9e50277b6317f98584b6b547e73523043e34f018eecae767846805`
AWS Agente PCS 1.1.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz`	`bef078bf60a6d8ecde2e6c49cd34d088703f02550279e3bf483d57a235334dc6`
AWS Agente PCS 1.1.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.0-1.tar.gz`	`594c32194c71bccc5d66e5213213ae38dd2c6d2f9a950bb01accea0bbab0873a`

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
AWS Agente PCS 1.0.1-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.1-1.tar.gz`	`04e22264019837e3f42d8346daf5886eaacecd21571742eb505ea8911786bcb2`
AWS Agente PCS 1.0.0-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz`	`d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042144b134ce0`

Instalador do Slurm

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
Slurm 25.05.4-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz`	`3b0f93bce441d4f4f6935175f2c1e81cd961cb923adb416fa6689f5592047a7d`
Slurm 25.05.3-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.3-1.tar.gz`	`851bb5815b6700ceb30cc4a3fda204ca8ce362c14528c339908983255a936cf0`
Slurm 24.11.6-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs`	`f17cd78e0bc6b9c818b794d9d2685cceabdc`

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
	-slurm-24.11-insta ller-24.11.6-2.tar.gz	73f4fbb12f7566ae5b 86a5abc32b
Slurm 24.11.6-1	https://aws-pcs-re po- *us-east-1* .s3.*us-* *east-1* .amazonaws.com/ aws-pcs-slurm/aws-pcs -slurm-24.11-insta ller-24.11.6-1.tar.gz	225de9fc18206f5f65 f412effe1fd457614a c97ee9822b3ff804a4 52b0fae522
Slurm 24.11.5-1	https://aws-pcs-re po- *us-east-1* .s3.*us-* *east-1* .amazonaws.com/ aws-pcs-slurm/aws-pcs -slurm-24.11-insta ller-24.11.5-1.tar.gz	593efe4d66bef2f3e4 6d5a382fb5a32f7a3c a2510bcf1b3c85739f 4f951810d5
Slurm 24.05.8-2	https://aws-pcs-re po- *us-east-1* .s3.*us-* *east-1* .amazonaws.com/ aws-pcs-slurm/aws-pcs -slurm-24.05-insta ller-24.05.8-2.tar.gz	c494b0b55c319a4c2f 3faf668c759d46c32c 4c7aa94ae97d941283 28fe95364b
Slurm 24.05.8-1	https://aws-pcs-re po- *us-east-1* .s3.*us-* *east-1* .amazonaws.com/ aws-pcs-slurm/aws-pcs -slurm-24.05-insta ller-24.05.8-1.tar.gz	210a43b376af082bba d640b2032655885790 c5dab0e6489cc327c7 310a375849

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
Slurm 24.05.7-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.7-1.tar.gz`	`0b5ed7c81195de2628c78f37c79e63fc4ae99132ca6b019b53a0d68792ee82c5`
Slurm 24.05.5-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz`	`7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b`
Slurm 23.11.10-4 (obsoleto)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-4.tar.gz`	`bb2d8c919c69dba38d14358f49c7f0427564c5dd4af85a1c9eca2c57ceeae29a`
Slurm 23.11.10-3 (obsoleto)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-3.tar.gz`	`488a10ee0fbd57ec0e0ff7ea708a9e3038fafdc025c6bb391c75c2e2a7852a00`

Installer (Instalador)	Faça download do URL	SHA256 soma de verificação
Slurm 23.11.10-2 (obsoleto)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-2.tar.gz`	`0bbe85423305c05987931168caf98da08a34c25f9eec0690e8e74de0b7bc8752`
Slurm 23.11.10-1 (obsoleto)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-1.tar.gz`	`27e8faa9980e92cdfd8cfdc71f937777f0934552ce61e33dac4ecf5a20321e44`
Slurm 23.11.9-1 (obsoleto)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz`	`1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8`

Notas de lançamento da amostra AWS PCS AMIs

AMIs para obter as versões principais suportadas mais recentes do agendador, receba atualizações de segurança e correções críticas de bugs. Esses patches de segurança incrementais não estão incluídos nas notas oficiais de lançamento.

> **Important**
>
> Amostras AMIs relacionadas às versões antigas do agendador não são suportadas e não recebem atualizações.

> **Important**
>
> AMIs As amostras são para fins de demonstração e não são recomendadas para cargas de trabalho de produção.

Sumário

AWS Exemplo de PCS AMIs para x86_64 (Amazon Linux 2)

Fauna 25.05

Nome da AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador x86 de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por Architectures=x86_64.

Conteúdo da AMI

- Serviço da AWS compatível: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: x86_64
- Tipo de volume do EBS: gp2
- Instalador EFA: 1.43.1
- GDRCopy: 2.5.1
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Fauna 24.11

> **Note**
>
> AWS O PCS suporta a contabilização do Slurm 24.11 e versões posteriores. Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.

Nome da AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador x86 de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por`Architectures=x86_64`.

Conteúdo da AMI

- AWS Serviço suportado: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: x86_64
- Tipo de volume do EBS: gp2
- Instalador EFA: 1.33.0
- GDRCopy: 2,4
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Fauna 24.05

Nome da AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.05`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador x86 de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por`Architectures=x86_64`.

Conteúdo da AMI

- AWS Serviço suportado: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: x86_64
- Tipo de volume do EBS: gp2
- Instalador EFA: 1.33.0
- GDRCopy: 2,4
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Fauna 23.11

Nome da AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador x86 de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por`Architectures=x86_64`.

Conteúdo da AMI

- AWS Serviço suportado: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: x86_64
- Tipo de volume do EBS: gp2

- Instalador EFA: 1.33.0
- GDRCopy: 2,4
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

AWS Amostra de PCS AMIs para Arm64 (Amazon Linux 2)

Fauna 25.05

Nome da AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-25.05`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador Arm de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por Architectures=arm64.

Conteúdo da AMI

- Serviço da AWS compatível: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: arm64
- Tipo de volume do EBS: gp2
- Instalador EFA: 1.43.1
- GDRCopy: 2.5.1
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Fauna 24.11

> **Note**
>
> AWS O PCS suporta a contabilização do Slurm 24.11 e versões posteriores. Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.

Nome da AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.11`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador Arm de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por`Architectures=arm64`.

Conteúdo da AMI

- AWS Serviço suportado: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: arm64
- Tipo de volume do EBS: gp2
- Instalador EFA: 1.33.0
- GDRCopy: 2,4
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Fauna 24.05

Nome da AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.05`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador Arm de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por`Architectures=arm64`.

Conteúdo da AMI

- AWS Serviço suportado: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: arm64
- Tipo de volume do EBS: gp2
- Instalador EFA: 1.33.0
- GDRCopy: 2,4
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Fauna 23.11

Nome da AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-23.11`

Instâncias do EC2 com suporte

- Todas as instâncias com um processador Arm de 64 bits. Para encontrar instâncias compatíveis, navegue até o console do Amazon EC2. Escolha Tipos de instância e, em seguida, pesquise por`Architectures=arm64`.

Conteúdo da AMI

- AWS Serviço suportado: AWS PCS
- Sistema operacional: Amazon Linux 2
- Arquitetura de computação: arm64
- Tipo de volume do EBS: gp2

- Instalador EFA: 1.33.0
- GDRCopy: 2,4
- Driver NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Sistemas operacionais compatíveis no AWS PCS

AWS O PCS usa a Amazon Machine Image (AMI) configurada para um grupo de nós de computação para iniciar EC2 instâncias nesse grupo de nós de computação. A AMI determina o sistema operacional que as EC2 instâncias usam. Você não pode alterar o sistema operacional na amostra AWS PCS AMIs. Você deve criar uma AMI personalizada se quiser usar um sistema operacional diferente. Para obter mais informações, consulte Amazon Machine Images (AMIs) para AWS PCS.

Sistemas operacionais compatíveis

- Amazon Linux 2

  Esse é o sistema operacional na amostra AWS PCS AMIs.

  > **Important**
  >
  > AMIs As amostras são para fins de demonstração e não são recomendadas para cargas de trabalho de produção. Você deve criar e usar uma AMI personalizada para cargas de trabalho de produção, mesmo que pretenda usar o Amazon Linux 2.

- Amazon Linux 2023
- RedHat Linux empresarial 9 (RHEL 9)

  O custo sob demanda do RHEL de qualquer tipo de instância é maior do que para outros sistemas operacionais compatíveis. Para ter mais informações sobre preços, consulte Definição de preço sob demanda e How is Red Hat Enterprise Linux on Amazon Elastic Compute Cloud offered and priced?.

- RedHat Linux corporativo 8 (RHEL 8)
- Rocky Linux 9

  Você pode usar o Rocky Linux 9 oficial AMIs como base para uma AMI personalizada. Sua compilação personalizada da AMI pode falhar se a AMI básica não tiver o kernel mais recente.

  Para atualizar o kernel

  1. Execute uma instância usando um ID de AMI rocky9 aqui: https://rockylinux.org/cloud-images/
  2. ssh na instância e execute o seguinte comando:

```
sudo yum -y update
```

3. Crie uma imagem da instância. Você especifica essa imagem como a da `ParentImage` sua AMI personalizada.

- Rocky Linux 8
- Ubuntu 22.04

  O Ubuntu 22.04 requer chaves mais seguras para SSH e não suporta chaves RSA por padrão. Recomendamos que você gere e use uma ED25519 chave em vez disso.
- Ubuntu 24.04

AWS Versões do agente PCS

O software do agente AWS PCS configura as instâncias do EC2 que o AWS PCS lança para uso com o Slurm. Você inclui o agente em uma Amazon Machine Images (AMI) que você especifica ao criar grupos de nós de computação para seu cluster. As instâncias do EC2 lançadas nesses grupos de nós de computação usam a AMI especificada e o software agente AWS PCS incluído. O agente AWS PCS permite que uma instância do EC2 se registre como parte do cluster. Para usar o software de agente AWS PCS mais recente, você deve atualizar seu software personalizado AMIs. Para obter mais informações, consulte Etapa 2 — Instalar o agente AWS PCS em Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

AWS Versão do agente PCS	Data de lançamento	Notas da versão
v1.3.2-1	10 de março de 2026	• Corrigido um problema em que os nós de computaçã o executando o RHEL 8.10 ou o Rocky Linux 8.10 falhavam na inicialização devido a uma porta traseira SigV4 com defeito `curl` nesses sistemas operacion ais.
v1.3.1-1	7 de novembro de 2025	• Foi aprimorada a desativaç ão do hyperthreading usando o parâmetro sysfs `smt/control` quando disponível. • Corrigida uma possível condição de corrida quando a CPU é bloqueada durante a inicialização enquanto o agente PCS tenta desativar o hyperthreading. • Foi corrigido o problema que fazia com que os

AWS Versão do agente PCS	Data de lançamento	Notas da versão
		`InstanceType` campos `InstanceId` e dos nós de computação do Slurm fossem preenchidos com um timestamp e um hífen, respectivamente.
v1.3.0-1	3 de novembro de 2025	• Foi adicionado suporte para novos sistemas operacionais: Amazon Linux 2023, Ubuntu 24, RHEL 8, Rocky 8.
v1.2.2-1	16 de outubro de 2025	• Consultas de metadados de instância permitidas em um IPv6 endpoint se um IPv4 endpoint não estiver disponível. • Corrigido um problema que impedia que o hyperthreading fosse desativado se o kernel retornasse threads irmãos como intervalos de ID de CPU. • Correção de um problema que gerava falsas mensagens de falha nos registros quando o hyperthreading era desativado com sucesso.

AWS Versão do agente PCS	Data de lançamento	Notas da versão
v1.2.1-1	19 de junho de 2025	• O agente AWS PCS agora tenta iniciar o slurmd por até 30 minutos se o controlador não estiver disponível. • Corrigido um problema que produzia uma configura ção incorreta do slurmd se a resposta a RegisterC omputeNodeGroupInstance continha um endpoint do SLURMDBD.
v1.2.0-1	07 de março de 2025	• Suporte habilitado para IPv6 entradas`slurmd.conf` .
v1.1.1-1	13 de dezembro de 2024	• Corrigido um problema em que uma versão incorreta do Slurm era relatada na chamada para. RegisterC omputeNodeGroupInstance • Corrigido um problema em que os metadados da instância não eram buscados corretamente se um script personalizado `/opt/aws/pcs/etc/bootstrap_hooks/` fosse executado.
v1.1.0-1	06 de dezembro de 2024	• Habilitou `/opt/aws/pcs/etc/bootstrap_hooks/` a execução de scripts personalizados antes das etapas de bootstrap.

AWS Versão do agente PCS	Data de lançamento	Notas da versão
v1.0.1-1	22 de outubro de 2024	• Corrigido um problema em que os dispositivos NVIDIA não funcionavam quando `slurmd` iniciados em instâncias habilitadas para GPU.
v1.0.0-1	28 de agosto de 2024	• Versão inicial.

Programador Slurm no PCS AWS

O Slurm é um gerenciador de carga de trabalho de código aberto projetado para clusters Linux que fornece recursos de agendamento de tarefas, alocação de recursos e monitoramento de tarefas para cargas de trabalho de HPC. AWS O PCS oferece suporte ao agendador Slurm para gerenciar suas cargas de trabalho de cluster.

Tópicos

Versões Slurm no PCS AWS

O SchedMD aprimora continuamente o Slurm com novos recursos, otimizações e patches de segurança. O SchedMD lança uma nova versão principal em intervalos regulares e planeja oferecer suporte a até 3 versões a qualquer momento. AWS O PCS foi projetado para atualizar automaticamente o controlador Slurm com versões de patch.

Quando o SchedMD encerra o suporte para uma versão principal específica, o AWS PCS designa essa versão como End of Life (EOL). Após o EOL, nenhum novo cluster pode ser criado com essa versão, embora os clusters existentes possam continuar funcionando por até 12 meses sem suporte garantido. AWS O PCS envia um aviso prévio se uma versão principal do Slurm estiver próxima do EOL, para ajudar os clientes a saberem quando atualizar seus clusters para uma versão mais recente compatível.

Recomendamos que você use a versão mais recente compatível do Slurm para implantar seu cluster e acessar os avanços e melhorias mais recentes.

Versões do Slurm suportadas no PCS AWS

A tabela a seguir mostra as versões suportadas do Slurm e as datas e informações importantes de cada versão.

Versão Slurm	Data de lançamento do SchedMD	AWS Data de lançamento do PCS	AWS Data de EOL do PCS	Versão mínima compatível do agente AWS PCS	Amostra de AWS PCS compatível AMIs
25.05	29/05/2025	16/10/2025	31/05/2027	1.0.0-1	• `aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05` • `aws-pcs-sample_ami-amzn2-arm64-slurm-25.05`
24.11	29/11/2024	14/05/2025	31/05/2026	1.0.0-1	• `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11` • `aws-pcs-sample_ami`

Versão Slurm	Data de lançamento do SchedMD	AWS Data de lançamento do PCS	AWS Data de EOL do PCS	Versão mínima compatível do agente AWS PCS	Amostra de AWS PCS compatível AMIs
					`-amzn2-arm64-slurm-24.11`

Versões do Slurm não suportadas no PCS AWS

A tabela a seguir mostra as versões do Slurm que não são suportadas no AWS PCS.

Versão Slurm	Data de lançamento do SchedMD	AWS Data de lançamento do PCS	AWS Data de EOL do PCS		
24.05	30/05/2024	18/12/2024	30/11/2025		
23.11	21/11/2023	28/08/2024	31/05/2025		

Notas de lançamento das versões do Slurm no PCS AWS

Este tópico descreve mudanças importantes para cada versão do Slurm atualmente suportada no AWS PCS. Recomendamos que você analise as alterações entre a versão antiga e a nova ao atualizar seu cluster.

Fauna 25.05

Mudanças implementadas no AWS PCS

- O Slurm SchedulerParameter requeue_on_resume_failure agora está ativado por padrão.
- “stderr” foi removido como uma opção para LogTimeFormat, pois foi desativado no Slurm 25.05.
- AWS O PCS suporta a configuração de pacotes de vários clusters: o nó de login pode acessar vários clusters.

Para obter mais informações sobre o Slurm 25.05, consulte as seguintes publicações:

- Anúncio de lançamento do SchedMD: https://www.schedmd.com/slurm-version-25-05-0-is-now-available/
- Notas de lançamento do SchedMD: https://github.com/SchedMD/ slurm/blob/slurm-25-05-0-1/ RELEASE _NOTES.md

Fauna 24.11

Mudanças implementadas no AWS PCS

- AWS O PCS oferece suporte à contabilidade do Slurm. Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.

Para obter mais informações sobre o Slurm 24.11, consulte as seguintes publicações:

- Anúncio de lançamento do SchedMD
- Notas de lançamento do SchedMD

Fauna 24.05

Mudanças implementadas no AWS PCS

- O novo módulo Slurm Step Manager agora está habilitado por padrão no AWS PCS. Esse módulo oferece benefícios significativos ao transferir o gerenciamento de etapas do controlador central para os nós de computação, melhorando substancialmente a simultaneidade do sistema em ambientes com uso intenso de etapas. Para suportar essa configuração e melhor isolar `Prolog` e `Epilog` processar a execução, novos sinalizadores de prólogo (`Contain`,`Alloc`) são habilitados.
- A comunicação hierárquica do controlador para os nós de computação é habilitada para otimizar a comunicação entre nós do Slurm, o que melhora a escalabilidade e o desempenho. Além disso, a configuração de roteamento agora usa listas de nós de partição para comunicações do controlador, em vez do algoritmo de roteamento padrão do plug-in, aprimorando a resiliência do sistema.
- Um novo plugin de hash `HashPlugin=hash/sha3` substitui o anterior. `hash/k12 plugin` Agora, isso está habilitado por padrão nos clusters AWS PCS.

- Os registros do controlador Slurm agora incluem recursos aprimorados de auditoria para todas as chamadas de procedimento remoto (RPC) de entrada para. `slurmctld` Os registros incluem o endereço de origem, o usuário autenticado e o tipo de RPC antes do processamento da conexão.

Para obter mais informações sobre o Slurm 24.05, consulte as seguintes publicações:

- Anúncio de lançamento do SchedMD
- Notas de lançamento do SchedMD

Fauna 23.11

Configurações do Slurm que você pode alterar no PCS AWS

- O `SuspendTime` padrão é. `60` Use o parâmetro `scaleDownIdleTimeInSeconds` de configuração AWS PCS para defini-lo. Para obter mais informações, consulte o `scaleDownIdleTimeInSeconds`parâmetro do tipo de `ClusterSlurmConfiguration` dados na Referência da API AWS PCS.
- O `MaxJobCount` e `MaxArraySize` é baseado no tamanho que você escolher para o cluster. Para obter mais informações, consulte o `size`parâmetro da ação da `CreateCluster` API na Referência da API AWS PCS.
- A configuração do `SelectTypeParameters` Slurm é padronizada como. `CR_CPU` Você pode fornecê-lo como um valor `slurmCustomSettings` para defini-lo ao criar um cluster. Para obter mais informações, consulte o `slurmCustomSettings`parâmetro da ação da `CreateCluster` API e SlurmCustomSettingna Referência da API AWS PCS.
- Você pode definir `Prolog` e `Epilog` no nível do cluster. Você pode fornecê-lo como um valor `slurmCustomSettings` para defini-lo ao criar um cluster. Para obter mais informações, consulte `CreateCluster`e SlurmCustomSettingna Referência da API AWS PCS.
- Você pode definir `Weight` e `RealMemory` no nível do grupo de nós de computação. Você pode fornecê-lo como um valor `slurmCustomSettings` para defini-lo ao criar um grupo de nós de computação. Para obter mais informações, consulte `CreateComputeNodeGroup`e SlurmCustomSettingna Referência da API AWS PCS.

Perguntas frequentes sobre as versões do Slurm no PCS AWS

AWS O PCS mantém o suporte para várias versões do Slurm. Quando uma nova versão do Slurm é introduzida, o AWS PCS fornece suporte técnico e patches de segurança até que essa versão

chegue ao fim do suporte (EOS) do SchedMD. AWS PCS se refere à data EOS de uma versão do Slurm como fim da vida útil (EOL) para ser consistente com a terminologia. AWS

Por quanto tempo o AWS PCS suporta a versão Slurm?

AWS O suporte do PCS para as versões do Slurm está alinhado com os ciclos de suporte do SchedMD para as versões principais. AWS O PCS suporta a versão atual e as duas versões principais anteriores mais recentes. Quando o SchedMD lança uma nova versão principal, o AWS PCS encerra o suporte para a versão mais antiga suportada. AWS O PCS lança novas versões principais do Slurm o mais rápido possível, mas pode haver um atraso entre o lançamento do SchedMD e sua disponibilidade no PCS. AWS

Como meus clusters obtêm novos lançamentos da versão de patch do Slurm?

Para resolver bugs e correções de segurança, o AWS PCS foi projetado para aplicar automaticamente patches aos controladores de cluster que são executados em contas internas de propriedade do serviço. Para instalar patches em instâncias do EC2 em sua Conta da AWS, atualize a Amazon Machine Image (AMI) para seus grupos de nós de computação e atualize os grupos de nós de computação para usar a AMI atualizada. Para obter mais informações, consulte Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

> **Note**
>
> Os controladores Slurm não estão disponíveis enquanto os atualizamos. Os trabalhos em execução não são afetados. Os trabalhos enviados antes que o controlador do cluster fique indisponível são retidos até que o controlador esteja disponível.

Como sou informado sobre um próximo evento de EOL da versão Slurm?

Enviamos uma mensagem de e-mail 6 meses antes da data de EOL. Enviamos uma mensagem de e-mail a cada mês antes do EOL, com uma mensagem de e-mail final 1 semana antes da data do EOL. Após a data de EOL, enviamos mensagens de e-mail mensais por 12 meses para clientes que executam clusters AWS PCS com versões do EOL Slurm. Podemos suspender um cluster com uma versão do EOL Slurm se forem identificadas vulnerabilidades de segurança para essa versão.

Como posso determinar se a versão do Slurm usada pelo meu cluster está executando uma versão do EOL Slurm?

Enviamos uma mensagem de e-mail para notificá-lo de que você tem um cluster em execução com uma versão do EOL Slurm. Publicamos um alerta nos AWS Health Dashboard alertas que contém os detalhes de seus clusters com as versões do EOL Slurm. Você também pode usar o console AWS PCS para identificar os clusters com versões do EOL Slurm.

O que devo fazer se minha versão do Slurm estiver próxima ou além do EOL?

Crie um novo cluster com uma versão mais recente compatível do Slurm e atualize a versão do Slurm nas AMIs do seu grupo de nós de computação. A versão do Slurm em suas AMIs e instâncias do EC2 em execução não pode estar mais do que duas versões atrás da versão do Slurm do cluster. Para obter mais informações, consulte Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

O que acontecerá se eu não mudar para uma versão mais recente do Slurm até a data de EOL?

Você não pode criar novos clusters com uma versão do EOL Slurm. Os clusters existentes podem operar por até 12 meses sem AWS suporte, e nenhuma ação imediata é necessária para manter sua operação. Após a data de EOL, o suporte, as atualizações de segurança e a disponibilidade não são garantidos. Podemos suspender um cluster por motivos de segurança. É altamente recomendável que você use uma versão compatível do Slurm para manter a segurança e o suporte de seus clusters AWS PCS.

Quais são os riscos de operar um cluster com versões do EOL Slurm?

Clusters com versões do EOL Slurm apresentam riscos operacionais e de segurança significativos. Sem o monitoramento ativo do SchedMD, as vulnerabilidades de segurança podem permanecer sem serem detectadas ou resolvidas. Se vulnerabilidades críticas forem descobertas, poderemos suspender seus clusters imediatamente.

O que acontece com meus trabalhos, recursos de computação, armazenamento e rede do cluster quando meu cluster é suspenso?

Todos os recursos gerenciados pelo AWS PCS são encerrados. Isso inclui o controlador Slurm, grupos de nós de computação e instâncias do EC2. Todos os trabalhos executados em instâncias de computação são encerrados imediatamente e o cluster entra em um estado suspenso. Os recursos gerenciados pelo cliente, como sistemas de arquivos externos, permanecem intactos. Você pode usar o console AWS PCS e as ações da API para acessar a configuração do cluster.

Posso reiniciar um cluster suspenso para retomar seus trabalhos restantes?

Não, você não pode reiniciar um cluster suspenso. Você pode usar a configuração do cluster suspenso para criar um novo cluster com uma versão compatível do Slurm. Você pode executar os trabalhos restantes se os salvou em um sistema de arquivos externo.

Posso solicitar uma prorrogação além do período de carência de 12 meses?

Não, você não pode solicitar uma extensão para executar seu cluster além do período de carência de 12 meses. Oferecemos um prazo estendido para ajudá-lo a mudar para uma versão compatível do Slurm. Para evitar interrupções nas operações do cluster, recomendamos que você alterne antes que sua versão do Slurm atinja o EOL.

Contabilidade de slurm no PCS AWS

Você pode habilitar a contabilização em seus novos clusters AWS PCS para monitorar o uso do cluster, impor limites de recursos e gerenciar um controle de acesso refinado a filas específicas ou grupos de nós de computação. AWS O PCS cria e gerencia o banco de dados contábil do seu cluster, eliminando a necessidade de criar e gerenciar seu próprio banco de dados contábil separado. AWS O PCS usa o recurso de contabilidade no Slurm. Para obter mais informações sobre o recurso de contabilidade no Slurm, consulte a documentação do Slurm em SchedMD.

Para usar a contabilidade, ative-a ao criar um novo cluster e, opcionalmente, definir parâmetros contábeis. Depois que o status do cluster for `Active` e tiver grupos de nós de computação, você poderá se conectar ao shell Linux de um nó de login para realizar funções contábeis, como visualizar dados do trabalho com o comando Slurm`sacct`.

> **Note**
>
> A contabilidade é compatível com o Slurm 24.11 ou posterior.

AWS PCS console

Na página Criar cluster, você deve selecionar uma versão válida do Slurm (versão 24.11 ou posterior). Em Configurações do Agendador, habilite Contabilidade.

AWS PCS API

Forneça a `accounting` configuração em sua chamada para a ação `CreateCluster` da API. No `accounting` objeto, defina o `mode` para`STANDARD`. Para obter mais informações, consulte CreateClusterContabilidade na Referência da API AWS PCS.

O exemplo a seguir usa o AWS CLI para chamar a ação `CreateCluster` da API. A substring do valor do parâmetro permite a `accounting='{mode=STANDARD}'` contabilização.

```
aws pcs create-cluster --cluster-name cluster-name \
                       --scheduler type=SLURM,version=24.11 \
                       --size SMALL \
                       --networking subnetIds=cluster-subnet-
id,securityGroupIds=cluster-security-group-id \
                       --slurm-configuration
 scaleDownIdleTimeInSeconds=180,accounting='{mode=STANDARD}',slurmCustomSettings='[{paramete
```

> **Important**
>
> Você receberá cobranças adicionais se ativar a contabilidade. Para obter mais informações, consulte a página de preços do AWS PCS.

Modificando as configurações contábeis

Você pode ativar ou desativar a contabilização em clusters existentes sem reconstruir sua infraestrutura. Para obter mais informações, consulte Atualizando um cluster no AWS PCS.

Quando você desativa a contabilidade, o faturamento do recurso contábil é interrompido assim que o cluster entra no `UPDATING` estado. Quando você ativa a contabilidade, o faturamento começa quando o cluster retorna com sucesso ao `ACTIVE` estado.

Conceitos-chave para contabilidade Slurm no PCS AWS

Os conceitos a seguir são específicos do AWS PCS e controlam como AWS o PCS implementa a contabilidade do Slurm.

Banco de dados de contabilidade

AWS O PCS armazena seus dados contábeis em um banco de dados criado em um banco de dados Conta da AWS que AWS possui. Você não tem acesso ao `slurmdbd.conf`.

Tempo de purga padrão

Essa configuração de AWS PCS especifica o período de retenção (em dias) para todos os tipos de registros contábeis (trabalhos, eventos, reservas, etapas, suspensões, transações, dados de

uso). Por exemplo, se o valor for 30, o AWS PCS retém os registros contábeis por 30 dias. Você fornece esse valor ao criar o cluster. Se você não fornecer um valor, o AWS PCS reterá os registros contábeis no banco de dados indefinidamente.

AWS PCS console

Você especifica o tempo de limpeza padrão como parte das etapas para criar um cluster. Na página Criar cluster, você deve selecionar uma versão válida do Slurm (versão 24.11 ou posterior) e ativar a contabilização. Em Configurações do Agendador, forneça um valor inteiro para o tempo de limpeza padrão (dias).

AWS PCS API

Especifique o `defaultPurgeTimeInDays` como parte das `accounting` informações que você fornece em sua chamada para a ação da `CreateCluster` API. Para obter mais informações, consulte CreateClusterContabilidade na Referência da API AWS PCS.

> **Note**
>
> Quando você usa a API AWS PCS para criar um cluster, o valor padrão para `defaultPurgeTimeInDays` é `-1` e `0` não é um valor válido.

Aplicação da política contábil

Essa configuração determina com que rigor o Slurm aplica as regras de envio de trabalhos, os limites de recursos e as políticas contábeis para seu cluster. Essa configuração corresponde ao `AccountingStorageEnforce` parâmetro no `slurm.conf` arquivo do seu cluster. Você pode selecionar qualquer combinação de opções de fiscalização. Se você não selecionar nenhuma opção, não haverá restrições contábeis aplicadas aos trabalhos no cluster. AWS O PCS suporta as seguintes opções:

- associações — job-to-account mapeamento
- limites — restrições de recursos
- QoS — requisitos de qualidade de serviço
- modo de segurança — conclusão garantida dentro dos limites
- nosteps — desativa a contabilização de etapas
- nojobs — desativa a contabilização de tarefas

Para obter mais informações sobre essas opções, consulte a documentação do Slurm em SchedMD.

AWS PCS console

Você define as opções como parte das etapas para criar um cluster. Na página Criar cluster, você deve selecionar uma versão válida do Slurm (versão 24.11 ou posterior) e ativar a contabilização. Selecione as opções desejadas na lista suspensa Aplicação da política contábil em Configurações do Agendador.

AWS PCS API

No Slurm, essas opções são definidas no arquivo de um cluster. `slurm.conf` Você não tem acesso direto ao `slurm.conf` para seu cluster AWS PCS. Em vez disso, você fornece `SlurmCustomSettings` à `CreateCluster` API a ação ao criar um cluster. Para obter mais informações, consulte CreateClustera Referência da API AWS PCS.

Obtenha a configuração contábil para um cluster AWS PCS existente

A configuração de contabilidade do Slurm está incluída na configuração do Slurm do seu cluster.

AWS PCS console

1. Escolha Clusters no painel de navegação.
2. Escolha o nome do cluster na lista.
3. Na guia Configuração, encontre a configuração contábil em Configuração do Slurm

AWS PCS API

Use a ação `GetCluster` da API para obter a configuração do cluster. Você pode encontrar a configuração contábil no`slurmConfiguration`. A configuração para `mode` e o valor de `defaultPurgeTimeInDays` estão abaixo`accounting`. As opções selecionadas de aplicação da política contábil estão em`slurmCustomSettings`. Para obter mais informações, consulte GetClustera Referência da API AWS PCS.

API REST do Slurm em PCS AWS

AWS O PCS fornece suporte gerenciado para a API REST nativa do Slurm`slurmrestd`, fornecendo uma interface HTTP para interação programática com clusters. Você pode enviar trabalhos,

monitorar o status do cluster e gerenciar recursos por meio de solicitações HTTP padrão sem precisar de acesso direto ao shell ao seu cluster.

Casos de uso comuns

A API REST do Slurm oferece suporte a vários cenários de integração:

- Integração de aplicativos da Web: crie front-ends e aplicativos da web personalizados que enviam e gerenciam trabalhos diretamente.
- Integração com o Jupyter Notebook: permite que os pesquisadores enviem trabalhos de ambientes de notebook sem sair do fluxo de trabalho de desenvolvimento.
- Integração de soluções de parceiros: conecte ferramentas de HPC e gerenciadores de fluxo de trabalho de terceiros aos seus clusters de AWS PCS.
- Gerenciamento programático de clusters: automatize os fluxos de trabalho de envio, monitoramento e gerenciamento de recursos de tarefas.
- Fluxos de trabalho de computação de pesquisa: Support ambientes de pesquisa acadêmica e empresarial que exigem gerenciamento de tarefas orientado por API.

Requisitos e limitações

Antes de usar a API REST do Slurm, revise estes detalhes:

- Seu cluster deve usar a versão 25.05 ou superior do Slurm.
- O endpoint da API só poderá ser acessado por meio de um endereço IP privado na VPC do seu cluster.
- Seu grupo de segurança do cluster deve permitir tráfego HTTP na porta 6820.
- A autenticação requer tokens JWT com declarações de identidade de usuário específicas.

As limitações atuais incluem:

- Os tokens gerados por não `scontrol token` são suportados.
- `X-SLURM-USER-NAME`a representação do cabeçalho não está disponível.
- Algumas funcionalidades exigem que a contabilidade do Slurm esteja ativada.
- Não é compatível com o mecanismo do plug-in de filtro CLI do Slurm.

- As conexões com o endpoint da API REST não são criptografadas com TLS.

Tópicos

Habilitando a API REST do Slurm no PCS AWS

Ative a API REST do Slurm para acessar a interface HTTP do seu cluster para gerenciamento e monitoramento programáticos de tarefas. Você pode ativar esse recurso durante a criação do cluster ou atualizar um cluster existente que atenda aos requisitos.

Pré-requisitos

Antes de ativar a API REST do Slurm, verifique se você tem:

- Versão do cluster: Slurm versão 25.05 ou superior.
- Grupo de segurança: regras que permitem o tráfego HTTP na porta 6820 a partir das fontes desejadas.

Procedimento

Para habilitar a API REST do Slurm em um novo cluster

Console de gerenciamento da AWS

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. Selecione Criar cluster.
3. Em Detalhes do cluster, escolha Slurm versão 25.05 ou superior.
4. Defina as outras configurações do cluster conforme necessário.
5. Na seção Configuração do Agendador, defina a API REST como Ativada.
6. Configure seu grupo de segurança de cluster para permitir tráfego HTTP na porta 6820 a partir das fontes desejadas.

7. Conclua o processo de criação do cluster.

AWS CLI

1. Adicione uma configuração REST do Slurm ao criar seu cluster.

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM, version=25.05 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1 \
    --slurm-configuration slurmRest='{mode=STANDARD}'
```

2. Configure seu grupo de segurança de cluster para permitir tráfego HTTP na porta 6820 a partir das fontes desejadas.

Para habilitar a API REST do Slurm em um cluster existente

Console de gerenciamento da AWS

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. Escolha seu cluster na lista.
3. Verifique se seu cluster usa a versão 25.05 ou superior do Slurm nos detalhes do cluster.
4. Escolha Editar cluster.
5. Na seção Configuração do Agendador, defina a API REST como Ativada.
6. Escolha Atualizar cluster para aplicar as alterações.
7. Configure seu grupo de segurança de cluster para permitir tráfego HTTP na porta 6820 a partir das fontes desejadas.

AWS CLI

1. Atualize seu cluster com uma configuração REST do Slurm, como neste exemplo.

```
aws pcs update-cluster --cluster-identifier my-cluster \
    --slurm-configuration 'slurmRest={mode=STANDARD}'
```

2. Configure seu grupo de segurança de cluster para permitir tráfego HTTP na porta 6820 a partir das fontes desejadas.

O que acontece depois de ativar

Quando você ativa a API REST, o AWS PCS automaticamente:

- Gera uma chave de assinatura do JWT e a armazena no AWS Secrets Manager.
- Expõe o endpoint da API `https://<clusterPrivateIpAddress>:6820` em sua VPC.
- Atualiza a configuração do cluster para mostrar os detalhes do endpoint da API REST.

Agora você pode autenticar e usar a API REST para gerenciamento de tarefas e operações de cluster.

Autenticação com a API REST do Slurm no PCS AWS

A API REST do Slurm no AWS PCS usa a autenticação JSON Web Token (JWT) para garantir acesso seguro aos recursos do cluster. AWS O PCS fornece uma chave de assinatura gerenciada armazenada no AWS Secrets Manager, que você usa para gerar tokens JWT contendo declarações de identidade de usuário específicas.

Pré-requisitos

Antes de se autenticar com a API REST do Slurm, verifique se você tem:

- Configuração de cluster: cluster AWS PCS com Slurm 25.05+ e API REST habilitada.
- Permissões da AWS: acesso ao AWS Secrets Manager para a chave de assinatura do JWT.
- Informações do usuário: nome de usuário, ID de usuário POSIX e um ou mais grupos POSIX IDs para sua conta de cluster.
- Acesso à rede: conectividade na VPC do seu cluster com o grupo de segurança que permite a porta 6820.

Procedimento

Para recuperar o endereço do endpoint da API Slurm REST

Console de gerenciamento da AWS

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. Escolha seu cluster na lista.

3. Nos detalhes da configuração do cluster, localize a seção Endpoints.
4. Observe o endereço IP privado e a porta da API REST do Slurm (slurmrestd).
5. Você pode fazer chamadas de API enviando solicitações HTTP formatadas corretamente para esse endereço.

AWS CLI

1. Consulte o status do seu cluster com`aws pcs get-cluster`. Procure o `SLURMRESTD` endpoint no `endpoints` campo na resposta. Exemplo:

```
"endpoints": [
      {
          "type": "SLURMCTLD",
          "privateIpAddress": "192.0.2.1",
          "port": "6817"
      },
      {
          "type": "SLURMRESTD",
          "privateIpAddress": "192.0.2.1",
          "port": "6820"
      }
  ]
```

2. Você pode fazer chamadas de API enviando solicitações HTTP formatadas corretamente para `http://<privateIpAddress>:<port>/`

Para recuperar a chave de assinatura do JWT

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. Escolha seu cluster na lista.
3. Nos detalhes da configuração do cluster, localize a seção Autenticação do Agendador.
4. Observe o ARN e a versão da chave JSON Web Token (JWT).
5. Use o AWS CLI para recuperar a chave de assinatura do Secrets Manager:

```
aws secretsmanager get-secret-value --secret-
id arn:aws:secretsmanager:region:account:secret:name --version-id version
```

Para gerar um token JWT

1. Crie um JWT com as seguintes reivindicações obrigatórias:
   - `exp`— Tempo de expiração em segundos desde 1970 para o JWT
   - `iat`— Tempo atual em segundos desde 1970
   - `sun`— O nome de usuário para autenticação
   - `uid`— O ID de usuário POSIX
   - `gid`— O ID do grupo POSIX
   - `id`— Propriedades adicionais de identidade POSIX
     - `gecos`— Campo de comentário do usuário, geralmente usado para armazenar um nome legível por humanos
     - `dir`— Diretório inicial do usuário
     - `shell`— Shell padrão do usuário
     - `gids`— Lista de grupos POSIX adicionais em IDs que o usuário está
2. Assine o JWT usando a chave de assinatura recuperada do Secrets Manager.
3. Defina um prazo de expiração apropriado para o token.

> **Note**
>
> Como alternativa à `sun` reivindicação, você pode fornecer qualquer um dos seguintes:
>
> - `username`
> - Um nome de campo personalizado que você define por `userclaimfield` meio do `AuthAltParameters Slurm custom settings`
> - Um `name` campo dentro da `id` reivindicação

Para autenticar solicitações de API

1. Inclua o token JWT em suas solicitações HTTP usando um destes métodos:
   - Token do portador — Adicionar `Authorization: Bearer` *`<jwt>`* cabeçalho
   - Cabeçalho do Slurm — Adicionar cabeçalho `X-SLURM-USER-TOKEN:` *`<jwt>`*
2. Faça solicitações HTTP para o endpoint da API REST:

Aqui está um exemplo de como acessar a `/ping` API usando curl e o `Authorized: Bearer` cabeçalho.

```
curl -X GET -H "Authorization: Bearer <jwt>" \
      http://<privateIpAddress>:6820/slurm/v0.0.43/ping
```

Exemplo de geração de JWT

Obtenha a chave de assinatura JWT do cluster AWS PCS e armazene-a como um arquivo local. Substitua os valores de aws-region, secret-arn e secret version por valores apropriados para seu cluster.

```
#!/bin/bash
SECRET_KEY=$(aws secretsmanager get-secret-value \
  --region aws-region \
  --secret-id secret-arn \
  --version-stage secret-version \
  --query 'SecretString' \
  --output text)
echo "$SECRET_KEY" | base64 --decode > jwt.key
```

Este exemplo em Python ilustra como usar a chave de assinatura para gerar um token JWT:

```
#!/usr/bin/env python3

import sys
import os
import pprint
import json
import time
from datetime import datetime, timedelta, timezone
from jwt import JWT
from jwt.jwa import HS256
from jwt.jwk import jwk_from_dict
from jwt.utils import b64decode,b64encode
if len(sys.argv) != 3:
    sys.exit("Usage: gen_jwt.py [jwt_key_file] [expiration_time_seconds]")
SIGNING_KEY = sys.argv[1]
EXPIRATION_TIME = int(sys.argv[2])
with open(SIGNING_KEY, "rb") as f:
```

```
    priv_key = f.read()
signing_key = jwk_from_dict({
    'kty': 'oct',
    'k': b64encode(priv_key)
})
message = {
    "exp": int(time.time() + EXPIRATION_TIME),
    "iat": int(time.time()),
    "sun": "ec2-user",
    "uid": 1000,
    "gid": 1000,
    "id": {
        "gecos": "EC2 User",
        "dir": "/home/ec2-user",
        "gids": [1000],
        "shell": "/bin/bash"
    }
}
a = JWT()
compact_jws = a.encode(message, signing_key, alg='HS256')
print(compact_jws)
```

O script imprimirá um JWT na tela.

```
abcdefgtjwttoken...
```

Usando a API REST do Slurm para gerenciamento de tarefas no PCS AWS

Visão geral da API Slurm REST

A API REST do Slurm fornece acesso programático às funções de gerenciamento de cluster por meio de solicitações HTTP. A compreensão dessas características principais ajudará você a usar a API com o AWS PCS de forma eficaz:

- Protocolo de acesso: a API usa HTTP (não HTTPS) para comunicação na rede privada do seu cluster.
- Detalhes da conexão: acesse a API usando o endereço IP privado do seu cluster e a `slurmrestd` porta (normalmente 6820). O formato completo do URL base é`http://<privateIpAddress>:6820`.

- Controle de versão da API: a versão da API corresponde à sua instalação do Slurm. Para o Slurm 25.05, use a versão v0.0.43. O número da versão muda a cada lançamento do Slurm. Você pode encontrar as versões da API atualmente suportadas nas notas de lançamento do Slurm.
- Estrutura de URL: A estrutura de URL para a API REST do Slurm é. `http://<privateIpAddress>:<port>/<api-version>/<endpoint>` Informações detalhadas de uso dos endpoints da API REST podem ser encontradas na documentação do Slurm.

Pré-requisitos

Antes de usar a API REST do Slurm, certifique-se de ter:

- Configuração de cluster: cluster AWS PCS com Slurm 25.05+ e API REST habilitada.
- Autenticação: token JWT válido com declarações de identidade de usuário adequadas.
- Acesso à rede: conectividade na VPC do seu cluster com um grupo de segurança que permite a porta 6820.

Procedimento

Para enviar um trabalho usando a API REST

1. Crie uma solicitação de envio de trabalho com os parâmetros necessários:

```
{
  "job": {
    "name": "my-job",
    "partition": "compute",
    "nodes": 1,
    "tasks": 1,
    "script": "#!/bin/bash\necho 'Hello from Slurm REST API'"
  }
}
```

2. Envie o trabalho usando uma solicitação HTTP POST:

```
curl -X POST \
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '<job-json>' \
```

```
  https://<privateIpAddress>:6820/slurm/v0.0.43/job/submit
```

3. Anote o ID do trabalho retornado na resposta para fins de monitoramento.

Para monitorar o status do trabalho

1. Obtenha informações sobre um trabalho específico:

```
curl -X GET -H "Authorization: Bearer <jwt>" \
    https://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

2. Liste todos os trabalhos do usuário autenticado:

```
curl -X GET -H "Authorization: Bearer <jwt>" \
    https://<privateIpAddress>:6820/slurm/v0.0.43/jobs
```

Para cancelar um trabalho

- Envie uma solicitação DELETE para cancelar um trabalho específico:

```
curl -X DELETE -H "Authorization: Bearer <jwt>" \
    https://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

Perguntas frequentes sobre a API REST do Slurm no PCS AWS

Esta seção responde às perguntas mais frequentes sobre a API REST do Slurm no AWS PCS.

O que é a API REST do Slurm?

A API REST do Slurm é uma interface HTTP que permite interagir programaticamente com o gerenciador de carga de trabalho do Slurm. Você pode usar métodos HTTP padrão, como GET, POST e DELETE, para enviar trabalhos, monitorar o status do cluster e gerenciar recursos sem exigir acesso à linha de comando ao cluster.

Posso usar tokens gerados por**`scontrol token`**?

Não, a `scontrol token` saída padrão não é compatível com o AWS PCS. A API REST do PCS Slurm requer tokens JWT enriquecidos contendo declarações de identidade específicas

que incluem nome de usuário (`sun`), ID de usuário POSIX () e grupo (`uid`). IDs `gids` Os tokens Standard Slurm não possuem essas reivindicações obrigatórias e serão rejeitados pela API.

Posso acessar a API de fora da minha VPC?

Não, o endpoint da API REST só pode ser acessado de dentro da sua VPC usando o endereço IP privado do controlador Slurm. Para habilitar o acesso externo, implemente AWS serviços como o Application Load Balancer com VPC Link, API Gateway ou estabeleça conexões VPN ou emparelhamento de VPC para conectividade segura.

Por que a API usa HTTP em vez de HTTPS?

A API REST do Slurm foi projetada para ser um endpoint interno na rede privada do seu cluster. Para implantações de produção que exigem criptografia, você pode implementar a SSL/TLS terminação em um nível mais alto em sua arquitetura, como por meio de um gateway de API, balanceador de carga ou proxy reverso.

Como faço para controlar o acesso à API REST?

Configure as regras do grupo de segurança do seu cluster para restringir o acesso à porta 6820 no controlador Slurm. Defina regras de entrada para permitir conexões somente de intervalos de IP confiáveis ou fontes específicas em sua VPC, bloqueando o acesso não autorizado ao endpoint da API.

Como faço para girar a chave de assinatura do JWT?

Coloque seu cluster em modo de manutenção sem instâncias ativas e, em seguida, inicie a rotação de chaves por meio do AWS Secrets Manager. Após a conclusão da rotação, reative as filas. Todos os tokens JWT existentes se tornarão inválidos e deverão ser regenerados usando a nova chave de assinatura do Secrets Manager.

Preciso habilitar a contabilidade do Slurm para usar a API REST?

Não, a contabilidade do Slurm não é necessária para operações básicas da API REST, como envio e monitoramento de trabalhos. No entanto, todo o `/slurmdb` endpoint exige que a contabilidade esteja ativa.

Quais ferramentas de terceiros funcionam com a API REST do AWS PCS?

Muitos clientes existentes da API REST do Slurm devem trabalhar com AWS PCS, incluindo o Slurm Exporter for Prometheus, e aplicativos personalizados que sigam o formato padrão da API REST do Slurm. SlurmWeb No entanto, as ferramentas que dependem `scontrol token` da autenticação precisarão ser modificadas para funcionar com os requisitos do AWS PCS JWT.

Há algum custo adicional para usar a API REST?

Não, não há cobranças adicionais para ativar ou usar o recurso da API REST do Slurm. Você paga apenas pelos recursos do cluster subjacente, como de costume.

Como posso solucionar problemas com a API REST?

- Problemas de conectividade de rede

  Se você não conseguir acessar o endpoint da API, verá tempos limite de conexão ou erros de "conexão recusada" ao fazer solicitações HTTP ao controlador de cluster.

  O que fazer: verifique se seu cliente está na mesma VPC ou tem roteamento de rede adequado e confirme se seu grupo de segurança permite tráfego HTTP na porta 6820 do seu IP ou sub-rede de origem.

- Problemas de autenticação REST do Slurm

  Se seu token JWT for inválido, expirado ou assinado incorretamente, as solicitações de API retornarão "Erro de autenticação de protocolo" no campo de erros da resposta.

  Exemplo de mensagem de erro:

  ```
  {
  "errors": [
      {
      "description": "Batch job submission failed",
      "error_number": 1007,
      "error": "Protocol authentication error",
      "source": "slurm_submit_batch_job()"
      }
    ]
  }
  ```

  O que fazer: verifique se seu token JWT está formatado corretamente, não está expirado e assinado com a chave correta do Secrets Manager. Verifique se o token está formado corretamente e inclui as declarações necessárias e se você está usando o formato correto do cabeçalho de autenticação.

- Falha na execução do trabalho após o envio

  Se seu token JWT for válido, mas contiver conteúdo ou estrutura interna incorretos, os trabalhos podem ter entrado no estado paused (`PD`) com o código do motivo. `JobAdminHead`

Use `scontrol show job <job-id>` para inspecionar o trabalho — você verá `JobState=PENDING, Reason=JobHeldAdmin` e. `SystemComment=slurm_cred_create failure, holding job`

O que fazer: a causa raiz pode ser valores errados no JWT. Verifique se o token está estruturado adequadamente e inclui as declarações necessárias de acordo com a documentação do PCS.

- Problemas de permissão do diretório de trabalho

  Se a identidade do usuário especificada em seu JWT não tiver permissões de gravação no diretório de trabalho do trabalho, o trabalho falhará com erros de permissão, semelhantes ao uso `sbatch --chdir` com um diretório inacessível.

  O que fazer: certifique-se de que o usuário especificado em seu token JWT tenha as permissões apropriadas para o diretório de trabalho do trabalho.

Reinicializando nós de computação com o Slurm no PCS AWS

AWS O PCS suporta o comando nativo `scontrol reboot` do Slurm. Use esse comando para reinicializar os nós de computação sem a substituição da instância do EC2. Outros métodos de reinicialização (console do Amazon EC2 AWS CLI, patches automatizados ou manutenção do sistema) AWS fazem com que o PCS considere a instância do EC2 não íntegra e a substitua.

Benefícios da reinicialização do Slurm

A reinicialização do Slurm oferece várias vantagens para a manutenção do cluster:

- Preserve a capacidade — evite perder instâncias do EC2 com restrição de capacidade para outros clientes.
- Reduza os custos — elimine os ciclos desnecessários de substituição de instâncias e a cobrança contínua dos nós ociosos.
- Recuperação mais rápida — sem atrasos no provisionamento em comparação com a substituição da instância.
- Flexibilidade operacional — elimine vazamentos de memória, remova arquivos temporários e recupere nós de estados degradados.

Quando usar a reinicialização do Slurm

Use a reinicialização do Slurm para cenários comuns de manutenção operacional:

- Solução de problemas — Resolva problemas de desempenho ou processos que não respondem, especialmente para nós de GPU.
- Limpeza de recursos — Limpe vazamentos de memória, arquivos temporários ou processos bloqueados que afetam o desempenho do trabalho. `/tmp`
- Recuperação — recupere nós de estados paralisados ou degradados antes de exigir a substituição completa do nó.

Limitações

- Somente usuários do Slurm Admin (usuários root) podem executar comandos de reinicialização.
- O suporte de reinicialização é limitado a `scontrol reboot` apenas.
- RebootProgram a configuração não é suportada.
- Sem interface de console — somente linha de comando.

Tópicos

Reinicialize um nó de computação usando o Slurm no PCS AWS

Use o comando de reinicialização nativo do Slurm para resolver problemas de desempenho, eliminar problemas de recursos ou se recuperar de estados degradados sem perda da capacidade da instância EC2.

Pré-requisitos

- Privilégios de administrador do Slurm (acesso do usuário root)
- Acesso a um nó de login no cluster AWS PCS

Procedimento

1. Conecte-se a um nó de login por meio do console do EC2.
   a. No console do EC2, selecione Instances (Instâncias).
   b. Selecione sua instância do nó de login.
   c. Selecione Conectar.
2. Identifique o nome do nó de computação de destino usando `sinfo` ou`scontrol show node`.

```
sinfo
# or
scontrol show node
```

3. Execute o comando de reinicialização usando uma das seguintes opções:

> **Warning**
>
> Não use `nextstate=DOWN` com o `scontrol reboot` comando. Esse parâmetro marca o nó como não íntegro e aciona a substituição da instância.

- Reinicialização básica (espera que o nó fique ocioso):

```
scontrol reboot nodename
```

- Reinicialização imediata (drena o nó e reinicia quando os trabalhos são concluídos):

```
scontrol reboot ASAP nodename
```

- Reinicie com o motivo:

```
scontrol reboot ASAP reason="troubleshooting" nodename
```

- Reinicialize com o estado de retomada:

```
scontrol reboot ASAP nextstate=RESUME nodename
```

4. Monitore o progresso da reinicialização usando o. `scontrol show node`

```
scontrol show node nodename
```

5. Verifique se o nó volta ao serviço após a conclusão da reinicialização.

Cancelar uma reinicialização pendente no AWS PCS

Cancele uma reinicialização pendente para evitar tempo de inatividade desnecessário quando o problema for resolvido ou quando a reinicialização não for mais necessária.

Pré-requisitos

- Privilégios de administrador do Slurm
- O nó deve ter uma reinicialização pendente (mostrando o status “reinicialização emitida”)
- Acesso ao nó de login para execução de comandos

Procedimento

1. Conecte-se ao nó de login.
2. Verifique se o nó tem uma reinicialização pendente usando`scontrol show node`.

   ```
   scontrol show node nodename
   ```

   Procure por “reinicialização emitida” no status do nó.
3. Execute o comando cancel.

   ```
   scontrol cancel_reboot nodename
   ```
4. Verifique o cancelamento da reinicialização e o retorno do status do nó ao normal.

   ```
   scontrol show node nodename
   ```

Perguntas frequentes sobre a reinicialização do Slurm no PCS AWS

Encontre respostas para perguntas comuns sobre como usar a reinicialização do Slurm no PCS. AWS

O que é o suporte à reinicialização do Slurm?

Support para o comando nativo do Slurm. `scontrol reboot` Use esse comando para reinicializar os nós de computação sem a substituição automática da instância, o que preserva a capacidade da instância EC2 e reduz os custos operacionais.

Quem pode usar os comandos de reinicialização do Slurm?

Somente usuários do Slurm Admin (usuários root) podem executar comandos de reinicialização. Usuários comuns que tentarem usar `scontrol reboot` receberão um erro de permissão negada do Slurm sem afetar o nó.

O que acontece com os trabalhos em execução durante uma reinicialização?

Por padrão, os trabalhos são concluídos normalmente antes da reinicialização. Com a opção ASAP, o nó é drenado para evitar novos trabalhos, e a reinicialização ocorre após a conclusão dos trabalhos atuais. Os trabalhos podem ser cancelados ou colocados novamente na fila para reinicializações imediatas.

Como isso é diferente da reinicialização do console EC2?

A reinicialização do Slurm preserva a instância do EC2 e evita a substituição, enquanto as reinicializações do console do EC2 acionam o PCS para substituir a instância devido a falhas nas verificações de integridade durante o processo de reinicialização.

Posso configurar scripts de reinicialização personalizados?

Não, a RebootProgram configuração não é suportada na versão inicial. O recurso usa o comportamento padrão de reinicialização do Slurm sem suporte a scripts personalizados.

Quanto tempo demora a reinicialização do Slurm?

O tempo de reinicialização varia de acordo com o tipo de instância, os processos de inicialização do cliente, a configuração da AMI e se os trabalhos precisam ser concluídos primeiro. O processo inclui aguardar a conclusão dos trabalhos, reinicialização física, verificações de integridade e registro do daemon slurmd.

Posso ver um histórico de reinicializações?

Os eventos de reinicialização são registrados nos registros do Slurm (slurmctld e slurmd), que podem ser monitorados. CloudWatch O campo do motivo no status do nó mostra o motivo da reinicialização durante o processo.

E se um nó ficar preso durante a reinicialização?

Se um nó não concluir o processo de reinicialização interno ResumeTimeout, ele será marcado como INATIVO. Verifique se há erros CloudWatch nos registros, verifique a conectividade de rede e examine os registros do slurmd. Entre em contato com AWS o Support se os problemas persistirem.

Posso reinicializar vários nós ao mesmo tempo?

Sim, você pode especificar vários nós no comando de reinicialização:

```
scontrol reboot ASAP node1,node2,node3
```

Como posso reinicializar um nó sem esperar que os trabalhos sejam concluídos?

Para reinicializações imediatas dos nós ao enfrentar problemas como nós problemáticos que afetam tarefas de vários nós, degradação significativa do desempenho ou comportamento instável da GPU, você tem duas opções:

- Cancelar e reinicializar — Primeiro, cancele os trabalhos afetados usando e, em seguida`scancel <job_id>`, inicie uma reinicialização imediata usando. `scontrol reboot ASAP <nodename>` Os trabalhos em execução serão encerrados e precisarão ser reenviados após a recuperação do nó.
- Drenagem e reenfileiramento (menos impactante) — Comece iniciando uma drenagem e reinicie com`scontrol reboot ASAP <nodename>`, em seguida, reenfileire os trabalhos afetados usando. `scontrol requeue <job_id>` Isso coloca os trabalhos de volta ao estado pendente em vez de cancelá-los.

O que acontece se eu especificar nextState=DOWN?

Se você especificar`nextstate=DOWN`, o nó será marcado como não íntegro após a reinicialização e acionar a substituição da instância. Para evitar a substituição da instância, não especifique nextstate nem use. `nextstate=RESUME`

Recursos adicionais do

- Para obter os procedimentos básicos de reinicialização, consulteReinicialize um nó de computação usando o Slurm no PCS AWS.
- Para solucionar problemas de reinicialização, consulteSolucionando problemas de reinicialização do Slurm no PCS AWS.

- Para a documentação de reinicialização do Slurm, consulte a documentação do Slurm scontrol.

Solucionando problemas de reinicialização do Slurm no PCS AWS

Quando você encontrar problemas de reinicialização do nó, primeiro verifique o status do nó usando`scontrol show node` *`nodename`*. Em seguida, examine CloudWatch os registros do Slurm (slurmctld e slurmd) e dos registros do sistema para identificar possíveis erros.

Para solucionar problemas básicos, verifique a conectividade da rede, verifique as configurações do grupo de segurança e garanta que todos os serviços necessários estejam em execução após a reinicialização. Se os problemas persistirem após as etapas básicas de solução de problemas, entre em contato com o AWS Support. Ao entrar em contato com o suporte, forneça trechos de log relevantes, informações sobre o status do nó e um cronograma da tentativa de reinicialização para ajudar a acelerar o processo de resolução.

Recursos adicionais do

- Para monitorar instâncias AWS PCS usando CloudWatch, consulte Monitoramento de instâncias AWS PCS usando a Amazon CloudWatch.
- Para solução geral de problemas, consulteSolução de problemas no serviço de computação AWS paralela.
- Para obter a documentação do Slurm, consulte o Guia de solução de problemas do Slurm.

Definindo configurações personalizadas do Slurm no PCS AWS

Use configurações personalizadas do Slurm para definir parâmetros adicionais do Slurm em recursos de cluster, fila e grupo de nós de computação. Esta versão adiciona suporte às configurações do Slurm nos recursos do Queue, fornecendo controle granular sobre os comportamentos específicos da partição.

Benefícios das configurações personalizadas do Slurm

As configurações personalizadas do Slurm fornecem controle sofisticado sobre seu ambiente de HPC AWS baseado em PC. Você pode implementar uma contabilidade detalhada, aplicar controles de acesso e otimizar a execução da carga de trabalho por meio de quality-of-service configurações e políticas de preempção. Esses recursos garantem que trabalhos essenciais recebam os recursos

necessários e, ao mesmo tempo, mantêm a utilização eficiente do cluster. Se você gerencia cargas de trabalho aceleradas por GPU, implementa um agendamento de compartilhamento justo ou controla os ciclos de vida das tarefas, as configurações personalizadas ajudam a alinhar sua infraestrutura de HPC aos requisitos operacionais e aos objetivos da pesquisa.

Definindo configurações personalizadas

As configurações personalizadas do Slurm podem ser definidas por meio do AWS console, da CLI ou SDKs durante a criação do recurso ou modificadas posteriormente por meio de operações de atualização.

Console de gerenciamento da AWS

Navegue até Configurações adicionais do agendador na página de criação ou edição para qualquer tipo de recurso (cluster, fila ou grupo de nós de computação).

Para adicionar uma nova configuração

1. Escolha Adicionar nova configuração.
2. Selecione um nome de parâmetro na lista suspensa (que inclui breves descrições de parâmetros).
3. Forneça o valor correspondente.

Para cancelar a definição de uma configuração personalizada

1. Escolha Remover ao lado do parameter/value par relevante.
2. Crie ou atualize o recurso.

AWS CLI

Para gerenciamento programático de configurações personalizadas, use o `SlurmCustomSettings` campo nas operações de criação ou atualização.

Example— Atualizando o Prolog parâmetro em um cluster

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/prolog.sh"}]'
```

Example— Configurando uma fila para estar Default em um cluster

```
aws pcs update-queue \
    --cluster-identifier my-cluster \
    --queue-identifier my-queue \
    --slurm-configuration
 'SlurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Example— Configuração personalizada Features em um grupo de nós de computação

```
aws pcs update-compute-node-group \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-cng-1 \
    --slurm-configuration \
    'SlurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

Validação e tratamento de erros

AWS O PCS implementa um processo de validação em várias camadas para configurações personalizadas do Slurm. Durante as operações de criação e atualização, realizamos validações síncronas que incluem:

- Verificações em nível de campo: validamos configurações individuais para tipos de dados corretos, valores permitidos e requisitos de formato. Por exemplo, garantimos que os valores de tempo estejam no formato correto do Slurm e que os valores booleanos usem representações booleanas aceitas do Slurm.
- Validações sensíveis ao contexto: algumas configurações são verificadas em relação ao contexto de configuração mais amplo. Por exemplo, certos parâmetros só são válidos quando a contabilidade do Slurm está ativada.
- Consistência entre configurações: verificamos se as opções mutuamente exclusivas não estão definidas juntas e se as configurações interdependentes estão definidas corretamente.

Se a validação falhar, você receberá um `ValidationException` código de erro específico (por exemplo, InvalidInput), uma mensagem de erro clara descrevendo o problema e uma lista dos campos inválidos e seus respectivos detalhes de erro.

Embora muitos problemas sejam detectados durante essa validação inicial, algumas interações complexas entre as configurações só podem se tornar aparentes ao aplicar a configuração. Nesses

casos, a operação falhará com uma mensagem de erro informativa e quaisquer alterações parciais serão revertidas.

Limitações

AWS O PCS implementa uma abordagem de lista de permissões para proteger a segurança do serviço e a estabilidade operacional. As configurações que podem comprometer a segurança da conta de serviço ou interferir nos recursos do serviço gerenciado são restritas. No entanto, avaliamos continuamente as necessidades dos clientes e podemos adicionar suporte para configurações adicionais com base nos comentários dos clientes.

Tópicos

- Configurações personalizadas do Slurm para AWS clusters PCS
- Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS
- Configurações personalizadas do Slurm para AWS filas PCS
- Solução de problemas de configurações personalizadas do Slurm no PCS AWS

Configurações personalizadas do Slurm para AWS clusters PCS

As seguintes configurações personalizadas do Slurm são suportadas no nível do cluster:

- AccountingStorageEnforce

  > **Important**
  >
  > AWS PCS suporta um subconjunto das opções para`AccountingStorageEnforce`. Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.

- AccountingStorageTRES
- AccountingStoreFlags
- DefMemPerCPU
- Epilog
- EnforcePartLimits
- FairShareDampeningFactor
- HealthCheckInterval
- HealthCheckNodeState

- HealthCheckProgram
- JobRequeue
- LaunchParameters
- Licenses
- MinJobAge

> **Note**
>
> AWS PCS suporta um valor mínimo de 5 segundos para`MinJobAge`.

- OverTimeLimit
- PreemptExemptTime
- PreemptMode
- PreemptParameters
- PreemptType
- PriorityCalcPeriod
- PriorityDecayHalfLife
- PriorityFavorSmall
- PriorityFlags
- PriorityMaxAge
- PriorityUsageResetPeriod
- PriorityWeightAge
- PriorityWeightAssoc
- PriorityWeightFairshare
- PriorityWeightJobSize
- PriorityWeightPartition
- PriorityWeightQOS
- PriorityWeightTRES
- PrivateData
- Prolog
- PrologFlags
- PropagatePrioProcess

- PropagateResourceLimits
- PropagateResourceLimitsExcept
- RequeueExit
- RequeueExitHold
- SchedulerParameters
- SelectTypeParameters
- SrunPortRange
- TaskEpilog
- TaskPluginParam
- TaskProlog
- UnkillableStepProgram
- UnkillableStepTimeout

Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS

As seguintes configurações personalizadas do Slurm são suportadas no nível do grupo de nós de computação:

- CpuSpecList
- Features
- MemSpecLimit
- RealMemory
- Weight

Configurações personalizadas do Slurm para AWS filas PCS

As seguintes configurações personalizadas do Slurm são suportadas no nível da fila:

- AllowAccounts
- AllowQoS
- Default
- DefaultTime

- DenyAccounts
- DenyQoS
- ExclusiveUser
- GraceTime
- MaxTime
- OverSubscribe
- OverTimeLimit
- PreemptMode
- PriorityJobFactor
- PriorityTier
- QOS
- TRESBillingWeights

Solução de problemas de configurações personalizadas do Slurm no PCS AWS

Se você encontrar erros ao criar ou atualizar recursos do AWS PCS com as configurações personalizadas do Slurm, poderá usar o registro para diagnosticar e resolver os problemas.

Solução de problemas de configurações personalizadas incompatíveis do Slurm

Problema: você recebe uma mensagem de erro semelhante à seguinte ao realizar operações de cluster, grupo de nós de computação ou fila:

```
{OPERATION} failed. The Slurm custom settings of the cluster might be incompatible.
 Check the settings and try again.
```

Esse erro pode ocorrer com as seguintes operações:

- CreateCluster
- CreateComputeNodeGroup
- UpdateComputeNodeGroup
- CreateQueue
- UpdateQueue

Solução: ative o registro para entender o problema específico e solucionar as configurações incompatíveis.

Para solucionar problemas de configurações personalizadas incompatíveis do Slurm

1. Crie o cluster, se ele ainda não existir, ou garanta que seu cluster existente esteja em um estado em que o registro possa ser ativado.
2. Ative o registro em seu cluster. Para obter instruções detalhadas, consulte Registro e monitoramento para AWS PCS.

   > **Note**
   >
   > O registro pode ser ativado quando o cluster estiver sendo criado.

3. Analise os registros para identificar o problema específico de configuração do Slurm que está causando a incompatibilidade.
4. Corrija as configurações personalizadas incompatíveis com base nas informações de registro e repita a operação.

Para obter informações sobre as configurações personalizadas do Slurm suportadas, consulte:

- Configurações personalizadas do Slurm para AWS clusters PCS
- Configurações personalizadas do Slurm para grupos de nós de computação AWS PCS
- Configurações personalizadas do Slurm para AWS filas PCS

Estenda a funcionalidade do Slurm no AWS PCS com plug-ins SPANK

Use os plug-ins SPANK (Slurm Plug-in Architecture for Node and job Kontrol) para estender e modificar o comportamento do Slurm durante a inicialização e execução de trabalhos em clusters PCS. AWS Os plug-ins SPANK fornecem uma interface genérica para interceptar e modificar os estágios de lançamento do trabalho.

Instale plug-ins SPANK na AMI do seu nó de computação e configure-os para personalizar o comportamento do seu cluster Slurm de acordo com seus requisitos de carga de trabalho. Para obter mais informações sobre o SPANK, consulte a documentação do SPANK no site do SchedMD.

Sumário

Instale plug-ins SPANK no AWS PCS

Siga a documentação do plug-in para instalar os plug-ins SPANK na sua AMI.

Compile plug-ins SPANK para a versão específica do Slurm em seu cluster. O instalador do Slurm fornecido pela AWS PCS armazena o Slurm em. `/opt/aws/pcs/scheduler/slurm-version` Ao compilar o plug-in, especifique a versão do Slurm.

O exemplo a seguir mostra como especificar a versão do Slurm para alguns plug-ins:

```
export CFLAGS="-I/opt/aws/pcs/scheduler/slurm-version/include"
```

Se você tiver várias versões do Slurm na AMI, compile o plug-in para cada versão. Armazene os plug-ins compilados em pastas versionadas.

O exemplo a seguir mostra como especificar a pasta de destino para alguns plug-ins:

```
export DESTDIR="your-preferred-versioned-path"
```

> **Important**
>
> Os plug-ins podem exigir variáveis diferentes. Veja a documentação oficial do plug-in que você está instalando.

Configurar plug-ins SPANK no AWS PCS

Por padrão, armazene os arquivos de configuração em`/etc/aws/pcs/scheduler/slurm-version/plugstack.conf.d/`.

Para armazenar sua configuração do SPANK em um local diferente, adicione seus locais a um arquivo de configuração no diretório padrão.

O exemplo a seguir mostra como incluir arquivos de configuração de outros diretórios:

```
# content of /etc/aws/pcs/scheduler/slurm-version/any-filename.conf
include path-to-your-configuration-folder/*.conf
include path-to-a-second-configuration-folder/*.conf
```

Armazene cada configuração em um arquivo dedicado ou em um arquivo comum. Você pode usar vários arquivos de configuração.

Os exemplos a seguir mostram exemplos de arquivos de configuração:

```
# content of path-to-your-or-default-config-folder/filename-1.conf
required path-to-plugin-1 arguments
optional path-to-plugin-2 arguments
```

```
# content of path-to-your-or-default-config-folder/filename-2.conf
required path-to-plugin-3 arguments
```

Para obter informações adicionais sobre como configurar seus plug-ins, consulte a documentação de configuração do SPANK no site do SchedMD.

Important

Defina as permissões da pasta para evitar alterações não autorizadas na configuração do seu plug-in.

Note

AWS O PCS não gerencia seus plug-ins SPANK. Se você receber erros relacionados a plug-ins, verifique os registros de erros em seus nós de computação.

Note

O Slurm registra incorretamente um erro semelhante ao seguinte ao carregar sua configuração do SPANK:

```
error: "Include" failed in file /etc/slurm/plugstack.conf line 3
```

Você pode ignorar esse erro. Isso não afeta o funcionamento dos plug-ins SPANK.

Perguntas frequentes sobre plug-ins SPANK no AWS PCS

Esta seção aborda perguntas comuns sobre a instalação e configuração de plug-ins SPANK em clusters AWS PCS.

Preciso instalar plug-ins SPANK nos nós de login e nos nós de computação?

Alguns plug-ins do SPANK não exigem instalação em todos os nós; mas para uma melhor compatibilidade, recomendamos que você instale todos os plug-ins do SPANK em cada nó.

Que configuração adicional é necessária para o uso em produção dos plug-ins SPANK?

Além da instalação e configuração básicas mostradas nos exemplos, as implantações de produção normalmente exigem configuração adicional. Plug-ins baseados em contêineres, como o Pyxis, podem exigir que você defina variáveis de ambiente para o Enroot, ative a PMI (Process Management Interface) e configure permissões para o tempo de execução do contêiner. Consulte a documentação específica do plug-in para obter os requisitos detalhados de implantação de produção.

Como soluciono problemas do plug-in SPANK?

AWS O PCS não gerencia plug-ins SPANK. Examine os registros de erros em seus nós de computação para solucionar problemas.

Use os plug-ins de filtro CLI do Slurm para personalizar o envio de trabalhos no PCS AWS

AWS O PCS suporta plug-ins de filtro CLI do Slurm para executar scripts Lua personalizados que validam e modificam os parâmetros de envio de trabalhos nos nós de login e computação. Para obter informações detalhadas sobre os plug-ins de filtro CLI, consulte a documentação da API do plug-in cli_filter no site do SchedMD.

Requisitos

Os plug-ins de filtro CLI exigem o Slurm versão 24.11 ou posterior e um script Lua implantado em todos os nós de login e computação.

> **Important**
>
> Para as versões 24.11 e 25.05 do Slurm, os plug-ins de filtro CLI exigem a instalação do Slurm AWS usando o instalador PCS Slurm (versão 24.11.6-2+ ou 25.05.4-1+). Para obter mais informações sobre a instalação do Slurm, consulte. Etapa 3 — Instalar o Slurm

Limitações e considerações de segurança

- Aplicação de segurança — Os plug-ins de filtro CLI podem ser facilmente ignorados por qualquer usuário e não devem ser usados para políticas críticas de segurança. Os usuários podem desativar os plug-ins de filtro CLI fornecendo uma configuração personalizada que foi `CLIFilterPlugins` desativada ao enviar trabalhos.
- Somente implementação de Lua — a implementação do script Lua é suportada. A implementação de C não é suportada.

Tópicos

Configurar plug-ins de filtro CLI do Slurm em um cluster PCS AWS

Configure os plug-ins de filtro CLI ao criar um novo cluster AWS PCS. Você pode ativar ou desativar os plug-ins de filtro CLI em clusters existentes usando a API de atualização ou o console sem recriar o cluster.

Pré-requisitos

Antes de configurar os plug-ins de filtro CLI, conclua estas tarefas:

- Escreva e teste um script Lua que implemente a API CLI Filter Plugin
- Nomeie seu script Lua com exatidão `cli_filter.lua`

- Escolha um método para implantar seu script em todas as instâncias do cluster (AMI, S3 ou sistema de arquivos)
- Verifique se você está usando o Slurm versão 24.11 ou posterior

Ativar plug-ins de filtro CLI em um novo cluster

AWS PCS console

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. No painel de navegação, escolha Clusters.
3. Selecione Criar cluster.
4. Selecione uma versão válida do Slurm (versão 24.11 ou posterior).
5. Em Configurações do agendador, expanda Configurações adicionais do agendador.
6. Adicione uma nova configuração personalizada do Slurm com o nome do parâmetro definido como `CliFilterPlugins` e o valor do parâmetro definido como. `cli_filter/lua`
7. Conclua a configuração restante do cluster e escolha Criar cluster.

AWS PCS API

Forneça a `slurmCustomSettings` configuração em sua chamada para a ação `CreateCluster` da API. Defina `parameterName` para `CliFilterPlugins` e `parameterValue` para`cli_filter/lua`. Para obter mais informações, consulte CreateClustera Referência da API AWS PCS.

O exemplo a seguir usa o AWS CLI para chamar a ação `CreateCluster` da API. A configuração personalizada `CliFilterPlugins=cli_filter/lua` ativa os plug-ins de filtro CLI.

```
aws pcs create-cluster --cluster-name cluster-name \
--scheduler type=SLURM,version=24.11 \
--size SMALL \
--networking subnetIds=cluster-subnet-id,securityGroupIds=cluster-security-group-id
 \
--slurm-configuration \
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/
lua"}]'
```

Implantar scripts de plug-in de filtro CLI

Para implantar scripts do CLI Filter Plugin em seu cluster

1. Certifique-se de que todos os AMIs usados em grupos de nós de computação tenham o Slurm instalado por meio do instalador do AWS PCS Slurm.

   > **Note**
   >
   > Se você usar o AWS PCS Sample AMI para todos os grupos de nós de computação, pule esta etapa. O Slurm já está instalado.

2. Implante seu `cli_filter.lua` script `/etc/aws/pcs/scheduler/slurm-<version>/cli_filter.lua` em todas as instâncias do cluster.

   Por exemplo, para a versão 24.11 do Slurm:

   ```
   /etc/aws/pcs/scheduler/slurm-24.11/cli_filter.lua
   ```

3. Inicie todos os nós de login e computação usando seu preparado AMIs.
4. Teste o envio do trabalho para verificar se o plug-in de filtro CLI está sendo executado corretamente.

Ativar ou desativar plug-ins de filtro CLI em clusters existentes

Você pode ativar ou desativar os plug-ins de filtro CLI em clusters existentes sem reconstruir sua infraestrutura. Para obter mais informações, consulte Atualizando um cluster no AWS PCS.

AWS PCS console

1. Abra o console AWS PCS em https://console.aws.amazon.com/pcs/.
2. No painel de navegação, escolha Clusters.
3. Selecione o cluster a ser atualizado.
4. Escolha Editar ação.
5. Na página Editar cluster, em Configurações adicionais do agendador:

- Para habilitar os plug-ins de filtro CLI: adicione uma nova configuração personalizada do Slurm com o nome do parâmetro definido como `CliFilterPlugins` e o valor do parâmetro definido como. `cli_filter/lua`
- Para desativar os plug-ins de filtro CLI: remova a configuração existente`CliFilterPlugins`.

6. Escolha Atualizar cluster para enviar as alterações.
7. Monitore o status do cluster, que aparece como “Atualizando” durante o processo e “Ativo” quando a atualização é concluída.

AWS PCS API

Use a ação da `UpdateCluster` API para ativar ou desativar os plug-ins de filtro CLI. Para obter mais informações, consulte UpdateClustera Referência da API AWS PCS.

Para habilitar plug-ins de filtro CLI em um cluster existente:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/
lua"}]'
```

Para desativar os plug-ins de filtro CLI em um cluster existente:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'slurmCustomSettings=[]'
```

Resultados esperados

Depois de concluir a configuração:

- Seu cluster é criado com o CLI Filter Plugin ativado
- Os envios de trabalhos acionam sua lógica de validação personalizada antes de chegar ao controlador Slurm
- Trabalhos não compatíveis são rejeitados com suas mensagens de erro personalizadas
- Os trabalhos compatíveis prosseguem normalmente por meio do agendador Slurm

Solução de problemas

Falta o script do plug-in de filtro CLI em qualquer nó

Sintomas: O envio do trabalho falha imediatamente com erro de carregamento do plug-in.

Causa provável: script não implantado em todas as instâncias ou nome ou caminho de arquivo incorreto.

Resolução: verifique se o script existe no caminho correto em todos os nós de login e computação com o nome de arquivo `cli_filter.lua` exato.

Configuração inválida do plug-in de filtro CLI

Sintomas: falha na criação do cluster com erro de validação.

Causa provável: `CliFilterPlugins` parâmetro não definido para `cli_filter/lua` formatar.

Resolução: Use o valor exato do parâmetro `cli_filter/lua` em`slurmCustomSettings`.

Use o Amazon S3 para implantar um script de plug-in de filtro CLI no PCS AWS

Use o S3 para implantar seu script do CLI Filter Plugin quando quiser atualizar a lógica de envio de trabalhos em um cluster ativo sem reconstruir. AMIs Essa abordagem baixa o script do S3 durante a execução da instância usando dados do usuário.

Pré-requisitos

Antes de implantar seu script usando o S3, conclua estas tarefas:

- Crie um bucket S3 com seu script Lua do CLI Filter Plugin
- Configure o perfil da instância do IAM com acesso de leitura ao bucket do S3
- Configure o endpoint do S3 VPC Gateway para acesso direto sem internet
- Prepare o script de dados do usuário para baixar do S3

Para implantar o script do plug-in de filtro CLI usando o S3

1. Faça o upload do seu `cli_filter.lua` script para o bucket do S3.

2. Configure seu perfil de instância do IAM com permissões de leitura do S3 para o bucket.
3. Adicione o código shell aos dados do usuário do seu modelo de lançamento para baixar o script:

```
aws s3 cp s3://my-bucket/cli_filter.lua /etc/aws/pcs/scheduler/slurm-24.11/
cli_filter.lua
chmod 644 /etc/aws/pcs/scheduler/slurm-24.11/cli_filter.lua
```

4. Implante grupos de nós de computação com seus modelos de lançamento atualizados.
5. Teste o envio do trabalho para verificar a funcionalidade do script.

Resultados esperados

Depois de concluir a implantação do S3:

- O script do plug-in de filtro CLI é baixado automaticamente para todas as instâncias durante a execução
- As atualizações de script no S3 são refletidas nas instâncias recém-lançadas
- As políticas de envio de trabalhos são aplicadas de forma consistente em todo o cluster

Solução de problemas

Acesso negado ao S3

Sintomas: falha na inicialização da instância ou o script não foi baixado.

Causa provável: falta de permissões do IAM ou do endpoint VPC S3.

Resolução: verifique se o perfil da instância do IAM tem `s3:GetObject` permissão e se o endpoint VPC S3 está configurado.

Traduza um script de plug-in do Slurm Job Submit para usar o CLI Filter Plugin no PCS AWS

Traduza seu script Lua existente do Job Submit Plugin para o CLI Filter Plugin ao migrar de outros ambientes do Slurm. O processo de tradução envolve a atualização de nomes de funções e padrões de acesso ao campo para trabalhar com a API do CLI Filter Plugin.

Pré-requisitos

Antes de traduzir seu script, conclua estas tarefas:

- Revise seu script Lua existente do Job Submit Plugin
- Entenda as diferenças entre o Job Submit e o CLI Filter Plugin APIs
- Acesse a documentação do plug-in de filtro CLI do Slurm

Para traduzir o script Job Submit Plugin para o CLI Filter Plugin

1. Revise suas funções de script existentes do Job Submit Plugin (`slurm_job_submit`,`slurm_job_modify`).
2. Identifique as funções equivalentes do CLI Filter Plugin:
   - `slurm_job_submit` se torna `slurm_cli_pre_submit`
   - Adicionar `slurm_cli_setup_defaults` para configuração de parâmetros padrão
   - Adicionar `slurm_cli_post_submit` para ações de pós-envio
3. Translate a lógica de validação de tarefas dos `job_desc` campos para o acesso ao `options` array:
   - `job_desc.account` se torna `options["account"]`
   - `job_desc.partition` se torna `options["partition"]`
   - `job_desc.features` se torna `options["constraint"]`
4. Atualize o registro de chamadas de `slurm.log_user()` para`slurm.log_error()`.
5. Teste seu script traduzido em um cluster de desenvolvimento.
6. Implante em seu cluster de produção seguindo o processo padrão de implantação do CLI Filter Plugin.

Resultados esperados

Depois de concluir a tradução:

- Seu script traduzido fornece uma validação equivalente ao envio de trabalhos
- Os usuários veem mensagens de erro e avisos semelhantes aos do seu Job Submit Plugin original
- As políticas de envio de trabalhos são mantidas durante a migração para o AWS PCS

Solução de problemas

Erros de tradução do script

Sintomas: Os envios de trabalhos falham com erros de execução de Lua.

Causa provável: acesso incorreto ao campo ou chamadas de função no script traduzido.

Resolução: revise a documentação da API do CLI Filter Plugin e compare os mapeamentos de campo entre as interfaces Job Submit e CLI Filter.

Perguntas frequentes sobre os plug-ins de filtro CLI do Slurm no PCS AWS

Analise essas perguntas frequentes sobre os plug-ins de filtro CLI.

Qual é a diferença entre o CLI Filter Plugin e o Job Submit Plugin?

O CLI Filter Plugin é executado no lado do cliente nos nós de login e computação antes que o envio do trabalho chegue ao controlador, enquanto o Job Submit Plugin é executado no lado do servidor no controlador após o envio do trabalho. O CLI Filter Plugin pode ser ignorado pelos usuários, mas não bloqueia o controlador, enquanto o Job Submit é seguro, mas pode afetar o desempenho do cluster durante a execução.

O AWS PCS é compatível com o plug-in Slurm Job Submit?

Não, o Job Submit Plugin não é suportado no AWS PCS. Em vez disso, use o CLI Filter Plugin para validação e modificação do envio de trabalhos.

Posso usar o plug-in de filtro CLI para fiscalizar a segurança?

Não, o CLI Filter Plugin pode ser ignorado por determinados usuários e não deve ser usado para fins de segurança. Use-o para aprimorar a experiência do usuário, definir parâmetros padrão e orientar políticas, em vez de políticas críticas de segurança.

Por que o script deve estar em todos os nós de computação, não apenas nos nós de login?

Comandos do Slurm `srun` podem ser executados em scripts de trabalho em nós de computação, o que também aciona a execução do CLI Filter Plugin. O script deve estar disponível sempre que os comandos do Slurm forem executados.

Posso modificar o script do CLI Filter Plugin em um cluster ativo?

Sim, se você usar a abordagem de implantação do S3 ou do sistema de arquivos. Novas instâncias receberão o script atualizado, mas as instâncias existentes precisam que o script seja atualizado manualmente ou por meio do método de implantação escolhido.

Posso usar scripts diferentes do CLI Filter Plugin em diferentes grupos de nós de computação?

Sim, mas isso não é recomendado. Você pode fornecer scripts com lógica diferente para diferentes grupos de nós de computação, mas é responsável por gerenciar interdependências e evitar a sobreposição de lógica. A maioria dos clientes fornece um conjunto de lógica em todo o cluster.

Posso usar o plug-in de filtro CLI com implementação C em vez de Lua?

A implementação de C não é suportada. Somente a implementação do script Lua é suportada no AWS PCS. O SchedMD recomenda que os clientes usem Lua em vez de C para facilitar o uso ao implementar plug-ins de filtro CLI.

Posso ativar ou desativar o plug-in de filtro CLI em um cluster existente?

Sim, você pode ativar ou desativar o plug-in de filtro CLI em clusters existentes usando a API de atualização sem recriar o cluster.

Solução de problemas do plug-in de filtro Slurm CLI no PCS AWS

Use essas informações de solução de problemas para resolver problemas comuns do CLI Filter Plugin.

O envio do trabalho falha imediatamente com erro de carregamento do plug-in

Sintomas: os usuários recebem mensagens de erro sobre o plug-in de filtro CLI ausente ou com falha ao enviar trabalhos.

Causas possíveis:

- O script do plug-in de filtro CLI está ausente em um ou mais nós
- Nome do arquivo de script incorreto (deve ser exatamente) `cli_filter.lua`
- Script implantado no caminho errado do diretório
- O script tem permissões de arquivo incorretas

Resolução:

- Verifique se o script existe `/etc/aws/pcs/scheduler/slurm-<version>/cli_filter.lua` em todos os nós de login e computação
- Verifique se o nome do arquivo do script é exatamente `cli_filter.lua`
- Certifique-se de que o script tenha permissões legíveis (644 ou similar)
- Teste a implantação do script em um único nó de login antes da implantação em todo o cluster

A criação do cluster falha com erro de validação do CLI Filter Plugin

Sintomas: A criação do cluster falha com um erro sobre um `CliFilterPlugins` parâmetro inválido.

Causas possíveis:

- Formato incorreto do valor do parâmetro em `slurmCustomSettings`
- Erro de digitação no nome ou valor do parâmetro

Resolução:

- Use o nome exato do parâmetro: `CliFilterPlugins`
- Use o valor exato do parâmetro: `cli_filter/lua`
- Verifique a sintaxe JSON na matriz `slurmCustomSettings`

O script do plug-in de filtro CLI é executado, mas a validação do trabalho não funciona conforme o esperado

Sintomas: os trabalhos são enviados com sucesso, mas a lógica de validação personalizada não aciona nem produz resultados inesperados.

Causas possíveis:

- Erros de sintaxe do script Lua
- Padrões de acesso de campo incorretos (usando a sintaxe do Job Submit Plugin em vez do CLI Filter Plugin)
- Erros lógicos nas condições de validação

Resolução:

- Verifique se há erros de sintaxe no script Lua
- Verifique se o acesso ao campo usa `options["field_name"]` formato em vez de `job_desc.field_name`
- Adicione instruções de registro para depurar o fluxo de execução do script

- Teste primeiro a lógica do script com casos de validação simples

Falha na implantação do script S3

Sintomas: as instâncias são iniciadas, mas o script do CLI Filter Plugin não é baixado do S3.

Causas possíveis:

- O perfil da instância do IAM não tem permissões de leitura do S3
- Endpoint VPC S3 não configurado
- Caminho incorreto do bucket ou objeto do S3 nos dados do usuário

Resolução:

- Verifique se o perfil da instância do IAM tem `s3:GetObject` permissão para seu bucket
- Configurar o endpoint do S3 VPC Gateway para acesso direto
- Verifique o nome do bucket S3 e o caminho do objeto no script de dados do usuário
- Analise os registros de dados do usuário da instância em busca de erros de download do S3

Segurança no serviço de computação AWS paralela

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O Modelo de Responsabilidade Compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam ao Serviço de Computação AWS Paralela, consulte AWS Serviços no escopo do programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS PCS. Os tópicos a seguir mostram como configurar o AWS PCS para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos de AWS PCS.

Tópicos

Proteção de dados no serviço de computação AWS paralela

O modelo de responsabilidade AWS compartilhada O modelo se aplica à proteção de dados no Serviço de Computação AWS Paralela. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para saber mais sobre a privacidade de dados, consulte as Data Privacy FAQ. Para saber mais sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou 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 AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o AWS PCS ou outro Serviços da AWS usando o console AWS CLI, a API ou AWS SDKs. Quaisquer dados inseridos em tags ou em 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, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Criptografia em repouso

A criptografia é ativada por padrão para dados em repouso quando você cria um cluster de Serviço de Computação AWS Paralela (AWS PCS) com a Console de gerenciamento da AWS AWS CLI,, API AWS PCS ou AWS SDKs. AWS O PCS usa uma AWS chave KMS própria para criptografar dados em repouso. Para obter mais informações, consulte Chaves do cliente e AWS chaves no Guia do AWS KMS desenvolvedor. Você também pode usar uma chave gerenciada pelo cliente. Para obter mais informações, consulte Política de chave KMS necessária para uso com volumes criptografados do EBS no PCS AWS.

O segredo do cluster é armazenado AWS Secrets Manager e criptografado com a chave KMS gerenciada pelo Secrets Manager. Para obter mais informações, consulte Trabalhando com segredos de cluster no AWS PCS.

Em um cluster AWS PCS, os seguintes dados estão em repouso:

- Estado do agendador — inclui dados sobre trabalhos em execução e nós provisionados no cluster. Esses são os dados nos quais o Slurm persiste, `StateSaveLocation` conforme definido em seu. `slurm.conf` Para obter mais informações, consulte a descrição StateSaveLocationna documentação do Slurm. AWS O PCS exclui os dados do trabalho após a conclusão de um trabalho.
- Segredo de autenticação do agendador — O AWS PCS o usa para autenticar todas as comunicações do agendador no cluster.

Para obter informações sobre o estado do agendador, o AWS PCS criptografa automaticamente os dados e os metadados antes de gravá-los no sistema de arquivos. O sistema de arquivos criptografados usa o algoritmo de criptografia AES-256 padrão do setor para dados em repouso.

Criptografia em trânsito

Suas conexões com a API AWS PCS usam criptografia TLS com o processo de assinatura Signature Version 4, independentemente de você usar o AWS Command Line Interface (AWS CLI) ou AWS SDKs. Para obter mais informações, consulte Assinatura de solicitações de AWS API no Guia AWS

Identity and Access Management do usuário. AWS gerencia o controle de acesso por meio da API com as políticas do IAM para as credenciais de segurança que você usa para se conectar.

AWS O PCS usa o TLS para se conectar a outros AWS serviços.

Em um cluster do Slurm, o agendador é configurado com o plug-in de autenticação que fornece `auth/slurm` autenticação para todas as comunicações do agendador. O Slurm não fornece criptografia no nível do aplicativo para suas comunicações. Todos os dados que fluem pelas instâncias do cluster permanecem locais na VPC do EC2 e, portanto, estão sujeitos à criptografia da VPC se essas instâncias oferecerem suporte à criptografia em trânsito. Para obter mais informações, consulte Criptografia em trânsito no Guia do usuário do Amazon Elastic Compute Cloud. A comunicação é criptografada entre o controlador (provisionado em uma conta de serviço) e os nós do cluster em sua conta.

Gerenciamento de chaves

AWS O PCS usa uma AWS chave KMS própria para criptografar dados. Para obter mais informações, consulte Chaves do cliente e AWS chaves no Guia do AWS KMS desenvolvedor. Você também pode usar uma chave gerenciada pelo cliente. Para obter mais informações, consulte Política de chave KMS necessária para uso com volumes criptografados do EBS no PCS AWS.

O segredo do cluster é armazenado AWS Secrets Manager e criptografado com a chave KMS gerenciada pelo Secrets Manager. Para obter mais informações, consulte Trabalhando com segredos de cluster no AWS PCS.

Privacidade do tráfego entre redes

AWS Os recursos de computação do PCS para um cluster residem em 1 VPC na conta do cliente. Portanto, todo o tráfego interno do serviço AWS PCS em um cluster permanece na AWS rede e não viaja pela Internet. A comunicação entre o usuário e os nós AWS PCS pode viajar pela Internet e recomendamos o uso de SSH ou Systems Manager para conectar-se aos nós. Para obter mais informações, consulte O que é AWS Systems Manager? no Guia do AWS Systems Manager usuário.

Você também pode usar as seguintes ofertas para conectar sua rede local a: AWS

- AWS Site-to-Site VPN. Para obter mais informações, consulte O que é AWS Site-to-Site VPN? no Guia do AWS Site-to-Site VPN usuário.
- Um AWS Direct Connect. Para obter mais informações, consulte O que é AWS Direct Connect? no Guia do AWS Direct Connect usuário.

Você acessa a API AWS PCS para realizar tarefas administrativas para o serviço. Você e seus usuários acessam as portas do endpoint do Slurm para interagir diretamente com o agendador.

Criptografia do tráfego da API

Para acessar a API AWS PCS, os clientes devem oferecer suporte ao Transport Layer Security (TLS) 1.2 ou posterior. Exigimos TLS 1.2 e recomendamos TLS 1.3. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos. Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Você também pode usar AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Criptografia do tráfego de dados

A criptografia de dados em trânsito é habilitada a partir de instâncias do EC2 suportadas que acessam o endpoint do agendador e entre ComputeNodeGroup instâncias de dentro do. Nuvem AWS Para obter mais informações, consulte Criptografia em trânsito.

Política de chave KMS necessária para uso com volumes criptografados do EBS no PCS AWS

AWS O PCS usa funções vinculadas ao serviço para delegar permissões a outras pessoas. Serviços da AWS A função vinculada ao serviço AWS PCS é predefinida e inclui as permissões que o AWS PCS exige para ligar para outras pessoas Serviços da AWS em seu nome. As permissões predefinidas também incluem acesso às suas chaves gerenciadas pelo cliente Chaves gerenciadas pela AWS , mas não às suas.

Este tópico descreve como configurar a política de chaves necessária para iniciar instâncias quando você especifica uma chave gerenciada pelo cliente para a criptografia do Amazon EBS.

> **Note**
>
> AWS O PCS não exige autorização adicional para usar o padrão Chave gerenciada pela AWS para proteger os volumes criptografados em sua conta.

Conteúdo

Visão geral do

Você pode usar o seguinte AWS KMS keys para a criptografia do Amazon EBS quando o AWS PCS inicia instâncias:

- Chave gerenciada pela AWS: uma chave de criptografia em sua conta que é criada por, pertencente a e gerenciada pelo Amazon EBS. Essa é a chave de criptografia padrão para uma nova conta. O Amazon EBS usa o Chave gerenciada pela AWS para criptografia, a menos que você especifique uma chave gerenciada pelo cliente.
- Chave gerenciada pelo cliente: uma chave de criptografia personalizada que você cria, possui e gerencia. Para obter mais informações, consulte Criar uma chave KMS no Guia do AWS Key Management Service desenvolvedor.

> **Note**
>
> A chave deve ser simétrica. O Amazon EBS não oferece suporte a chaves assimétricas gerenciadas pelo cliente.

Você configura as chaves gerenciadas pelo cliente ao criar instantâneos criptografados ou um modelo de execução que especifica volumes criptografados, ou quando opta por habilitar a criptografia por padrão.

Configurar políticas de chave

Suas chaves KMS devem ter uma política de chaves que permita ao AWS PCS iniciar instâncias com volumes do Amazon EBS criptografados com uma chave gerenciada pelo cliente.

Use os exemplos desta página para configurar uma política de chaves para dar ao AWS PCS acesso à sua chave gerenciada pelo cliente. Você pode modificar a política de chaves da chave gerenciada pelo cliente ao criar a chave ou posteriormente.

A política principal deve ter as seguintes declarações:

- Uma declaração que permite que a identidade do IAM especificada no `Principal` elemento use diretamente a chave gerenciada pelo cliente. Inclui permissões para realizar as `DescribeKey` operações AWS KMS `Encrypt` `DecryptReEncrypt*`,`GenerateDataKey*`,, e na chave.
- Uma declaração que permite que a identidade do IAM especificada no `Principal` elemento use a `CreateGrant` operação para gerar concessões que delegam um subconjunto de suas próprias permissões para aqueles Serviços da AWS que estão integrados com AWS KMS ou outro principal. Isso permite que eles usem a chave para criar recursos criptografados em seu nome.

Não altere nenhuma declaração existente na política ao adicionar as novas declarações de política à sua política principal.

Para obter mais informações, consulte:

- create-key na Referência de Comandos AWS CLI
- put-key-policy na AWS CLI Command Reference
- Encontre o ID da chave e o ARN da chave no Guia do desenvolvedor AWS Key Management Service
- Funções vinculadas a serviços para PCS AWS
- Criptografia do Amazon EBS no Guia do usuário do Amazon EBS
- AWS Key Management Service no Guia do desenvolvedor do AWS Key Management Service

Exemplo 1: seções da política de chaves que permitem acesso à chave gerenciada pelo cliente

Adicione as seguintes declarações de política à política principal da chave gerenciada pelo cliente. Substitua o ARN de exemplo pelo ARN da sua função vinculada ao serviço. `AWSServiceRoleForPCS` Este exemplo de política dá à função vinculada ao serviço AWS PCS (`AWSServiceRoleForPCS`) permissões para usar a chave gerenciada pelo cliente.

```
{
```

```
    "Sid": "Allow service-linked role use of the customer managed key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

```
{
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
        ]
    },
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": true
        }
     }
}
```

Exemplo 2: seções da política de chaves que permitem acesso entre contas à chave gerenciada pelo cliente

Se você criar uma chave gerenciada pelo cliente em uma conta diferente da do cluster AWS PCS, deverá usar uma concessão em combinação com a política de chaves para permitir o acesso entre contas à chave.

Para conceder acesso à chave

1. Adicione as seguintes declarações de política à política de chaves da chave gerenciada pelo cliente. Substitua o ARN de exemplo pelo ARN da outra conta. *111122223333*Substitua pela ID da conta real na Conta da AWS qual você deseja criar o cluster AWS PCS. Isso permite que você conceda permissão para que um usuário ou uma função do IAM na conta especificada crie uma concessão para a chave usando o seguinte comando da CLI. Por padrão, os usuários não têm acesso à chave.

```
{.
   "Sid": "Allow external account 111122223333 use of the customer managed key",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::111122223333:root"
       ]
   },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
   ],
   "Resource": "*"
}
```

```
{
   "Sid": "Allow attachment of persistent resources in external
 account 111122223333",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::111122223333:root"
```

```
        ]
    },
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": "*"
}
```

2. Na conta na qual você deseja criar o cluster AWS PCS, crie uma concessão que delegue as permissões relevantes à função vinculada ao serviço AWS PCS. O valor de `grantee-principal` é o ARN da função vinculada ao serviço. O valor de `key-id` é o ARN da chave.

   O exemplo a seguir do comando da CLI create-grant fornece à função vinculada ao serviço `AWSServiceRoleForPCS` nomeada na *111122223333* conta permissões para usar a chave gerenciada pelo cliente na conta. *444455556666*

   ```
   aws kms create-grant \
     --region us-west-2 \
     --key-id arn:aws:kms:us-
   west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
     --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
   pcs.amazonaws.com/AWSServiceRoleForPCS \
     --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
    "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"
   ```

   > **Note**
   >
   > O usuário que faz a solicitação deve ter permissões para usar a `kms:CreateGrant` ação.

   O exemplo de política do IAM a seguir permite que uma identidade do IAM (usuário ou função) na conta *111122223333* crie uma concessão para a chave gerenciada pelo cliente na conta*444455556666*.

   JSON

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

```
    {
      "Sid": "AllowCreationOfGrantForTheKMSKeyinExternalAccount444455556666",
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
  ]
}
```

Para obter mais informações sobre como criar uma concessão para uma chave do KMS em uma Conta da AWS diferente, consulte Concessões no AWS KMS no Guia do desenvolvedor do AWS Key Management Service .

> **Important**
>
> O nome do perfil vinculado ao serviço especificado como a entidade principal do beneficiário deve ser o nome de um perfil existente. Depois de criar a concessão, para garantir que a concessão permita que o AWS PCS use a chave KMS especificada, não exclua e recrie a função vinculada ao serviço.

Edite as principais políticas no AWS KMS console

Os exemplos nas seções anteriores mostram apenas como adicionar declarações a uma política de chaves, que é apenas uma maneira de alterar uma política de chaves. A maneira mais fácil de alterar uma política de chaves é usar a visualização padrão do AWS KMS console para políticas de chaves e tornar uma identidade (usuário ou função) do IAM um dos principais usuários da política de chaves apropriada. Para obter mais informações, consulte Usando a visualização Console de gerenciamento da AWS padrão no Guia do AWS Key Management Service desenvolvedor.

> **Warning**
>
> As declarações de política de visualização padrão do console incluem permissões para realizar AWS KMS `Revoke` operações na chave gerenciada pelo cliente. Se você revogar uma concessão que deu Conta da AWS acesso a uma chave gerenciada pelo cliente em sua conta, os usuários dessa chave Conta da AWS perderão o acesso aos dados criptografados e à chave.

Acesso Serviço de Computação Paralela da AWS usando um endpoint de interface ()AWS PrivateLink

Você pode usar AWS PrivateLink para criar uma conexão privada entre sua VPC e Serviço de Computação Paralela da AWS ()AWS PCS. Você pode acessar AWS PCS como se estivesse em sua VPC, sem o uso de um gateway de internet, dispositivo NAT, conexão VPN ou conexão. Direct Connect As instâncias na sua VPC não precisam de endereços IP públicos para acessar o AWS PCS.

Estabeleça essa conectividade privada criando um endpoint de interface, habilitado pelo AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar para o endpoint de interface. Estas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao AWS PCS.

Para obter mais informações, consulte Acesso Serviços da AWS por meio AWS PrivateLink do AWS PrivateLink Guia.

Considerações para AWS PCS

Antes de configurar um endpoint de interface para AWS PCS, consulte Acesse um serviço da AWS usando um endpoint VPC de interface no Guia.AWS PrivateLink

AWS PCS suporta fazer chamadas para todas as suas ações de API por meio do endpoint da interface.

Se sua VPC não tiver acesso direto à Internet, você deverá configurar um VPC endpoint para permitir que suas instâncias do grupo de nós de computação chamem a ação da API. AWS PCS `RegisterComputeNodeGroupInstance`

Crie um endpoint de interface para AWS PCS

Você pode criar um endpoint de interface para AWS PCS usar o console Amazon VPC ou AWS Command Line Interface o AWS CLI(). Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário do AWS PrivateLink .

Crie um endpoint de interface para AWS PCS usar o seguinte nome de serviço:

```
com.amazonaws.region.pcs
```

*region*Substitua pelo ID do Região da AWS para criar o endpoint, como`us-east-1`.

Se você habilitar o DNS privado para o endpoint da interface, poderá fazer solicitações de API a AWS PCS usando seu nome DNS regional padrão. Por exemplo, `.pcs.us-east-1.amazonaws.com`

Crie uma política de endpoint para seu endpoint de interface.

Uma política de endpoint é um recurso do IAM que pode ser anexado ao endpoint de interface. A política de endpoint padrão permite acesso total AWS PCS por meio do endpoint da interface. Para controlar o acesso AWS PCS permitido pela sua VPC, anexe uma política de endpoint personalizada ao endpoint da interface.

Uma política de endpoint especifica as seguintes informações:

- As entidades principais que podem realizar ações (Contas da AWS, usuários do IAM e perfis do IAM).
- As ações que podem ser realizadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte Controlar o acesso aos serviços usando políticas de endpoint no Guia do AWS PrivateLink .

Exemplo: política de VPC endpoint para ações AWS PCS

Veja a seguir um exemplo de uma política de endpoint personalizado. Quando você anexa essa política ao seu endpoint de interface, ela concede acesso às AWS PCS ações listadas para todos os principais do cluster com o especificado. *cluster-id region*Substitua pelo ID Região da AWS do cluster, como`us-east-1`. *account-id*Substitua pelo Conta da AWS número do cluster.

```
{
    "Statement": [
            {
                "Action": [
                "pcs:CreateCluster",
                "pcs:ListClusters",
                "pcs:DeleteCluster",
                "pcs:GetCluster",
                ],
                "Effect": "Allow",
```

```
                "Principal": "*",
                "Resource": [
                    "arn:aws:pcs:region:account-id:cluster/cluster-id*"
                ]
            }
        ]
}
```

Identity and Access Management for AWS Parallel Computing Service

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos do AWS PCS. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas de identidade e acesso ao serviço de computação AWS paralela).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como o serviço de computação AWS paralela funciona com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Exemplos de políticas baseadas em identidade para o serviço de computação AWS paralela)

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte O que é o IAM Identity Center? no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .

- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o serviço de computação AWS paralela funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao AWS PCS, saiba quais recursos do IAM estão disponíveis para uso com o AWS PCS.

Recursos do IAM que você pode usar com o AWS Parallel Computing Service

Recurso do IAM	AWS Suporte para PCS
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim

Recurso do IAM	AWS Suporte para PCS
Permissões de entidade principal	Sim
Perfis de serviço	Não
Perfis vinculados ao serviço	Sim

Para ter uma visão de alto nível de como o AWS PCS e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade para PCS AWS

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para PCS AWS

Para ver exemplos de políticas baseadas em identidade do AWS PCS, consulte. Exemplos de políticas baseadas em identidade para o serviço de computação AWS paralela

Políticas baseadas em recursos no PCS AWS

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico.

Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações políticas para AWS PCS

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do AWS PCS, consulte Ações definidas pelo serviço de computação AWS paralela na Referência de autorização de serviço.

As ações de política no AWS PCS usam o seguinte prefixo antes da ação:

```
pcs
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

```
"Action": [
      "pcs:action1",
      "pcs:action2"
         ]
```

Recursos de políticas para AWS PCS

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos do AWS PCS e seus ARNs, consulte Recursos definidos pelo serviço de computação AWS paralela na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo serviço de computação AWS paralela.

Para ver exemplos de políticas baseadas em identidade do AWS PCS, consulte. Exemplos de políticas baseadas em identidade para o serviço de computação AWS paralela

Chaves de condição de política para AWS PCS

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista das chaves de condição do AWS PCS, consulte Chaves de condição para o serviço de computação AWS paralela na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas pelo serviço de computação AWS paralela.

Para ver exemplos de políticas baseadas em identidade do AWS PCS, consulte. Exemplos de políticas baseadas em identidade para o serviço de computação AWS paralela

ACLs em AWS PCS

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com AWS PCS

Compatível com ABAC (tags em políticas): sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Usando credenciais temporárias com AWS o PCS

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Permissões principais entre serviços para AWS PCS

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer

solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Funções de serviço para AWS PCS

Compatível com perfis de serviço: não

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> Alterar as permissões de uma função de serviço pode interromper a funcionalidade do AWS PCS. Edite as funções de serviço somente quando o AWS PCS fornecer orientação para fazer isso.

Funções vinculadas a serviços para PCS AWS

Compatibilidade com perfis vinculados a serviços: sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas ao serviço AWS PCS, consulte. Funções vinculadas a serviços para PCS AWS

Exemplos de políticas baseadas em identidade para o serviço de computação AWS paralela

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos do AWS PCS. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo AWS PCS, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição para o serviço de computação AWS paralela na Referência de autorização de serviço.

Tópicos

- Práticas recomendadas de política
- Usando o console AWS PCS
- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do AWS PCS em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.

- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o console AWS PCS

Para acessar o console do AWS Parallel Computing Service, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do AWS PCS em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para obter mais informações sobre as permissões mínimas necessárias para usar o console AWS PCS, consultePermissões mínimas para AWS PCS.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

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

```
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS políticas gerenciadas para o Serviço de Computação AWS Paralela

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para saber mais, consulte AWS Políticas gerenciadas pela no Guia do usuário do IAM.

AWS política gerenciada: AWSPCSCompute NodePolicy

Você pode anexar AWSPCSCompute NodePolicy às suas entidades do IAM. Você pode anexar essa política a uma função do IAM do nó de computação do AWS PCS que você especifica para permitir que os nós que usam essa função se conectem a um cluster do AWS PCS.

AWS O PCS associa essa política a uma função de grupo de nós de computação quando você usa o console para criar um grupo de nós de computação.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `pcs:RegisterComputeNodeGroupInstance`— Permitir que um nó de computação AWS PCS (instância EC2) se registre em um cluster AWS PCS.

Para visualizar as permissões para esta política, consulte AWSPCSComputeNodePolicy na Referência de políticas gerenciadas pela AWS .

AWS política gerenciada: AWSPCSService RolePolicy

Você não pode se vincular AWSPCSService RolePolicy às suas entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que permite que o AWS PCS execute ações em seu nome. Para obter mais informações, consulte Funções vinculadas a serviços para PCS AWS.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ec2`— Permite que o AWS PCS crie e gerencie recursos do Amazon EC2.
- `iam`— Permite que a AWS PCS crie uma função vinculada a serviços para a frota do Amazon EC2 e transmita a função para o Amazon EC2.
- `cloudwatch`— Permite que a AWS PCS publique métricas de serviço na Amazon CloudWatch.
- `secretsmanager`— Permite que o AWS PCS gerencie segredos dos recursos do cluster AWS PCS.

Para visualizar as permissões para esta política, consulte AWSPCSServiceRolePolicy na Referência de políticas gerenciadas pela AWS .

AWS Atualizações do PCS para políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do AWS PCS desde que esse serviço começou a rastrear essas alterações. Para alertas automáticos sobre alterações nesta página, assine o feed RSS na página de histórico de documentos do AWS PCS.

Alteração	Descrição	Data
AWSPCSServiceRolePolicy: atualizar para uma política existente	AWS O PCS adicionou novas permissões para oferecer suporte a blocos de capacidad e para uma capacidade computacional previsível. Foi adicionada `ec2:Descr ibeCapacityReserva tions` permissão para	11 de setembro de 2025

Alteração	Descrição	Data
	permitir que o AWS PCS descubra e use reservas do Capacity Block para grupos de nós de computação.	
AWSPCSComputeNodePolicy – Nova política	AWS O PCS adicionou uma nova política para conceder permissão aos nós de computação do AWS PCS para se conectarem aos clusters do AWS PCS. AWS O PCS associa essa política a uma função do IAM quando você cria um grupo de nós de computação no console do AWS PCS.	23 de junho de 2025
Atualizou o JSON neste documento	Foi corrigido o JSON neste documento para incluir. `"arn:aws:ec2:*:*:spot-instances-request/*"`	5 de setembro de 2024
AWS O PCS começou a rastrear as mudanças	AWS A PCS começou a monitorar as mudanças em suas políticas AWS gerenciadas.	28 de agosto de 2024

Funções vinculadas a serviços para PCS AWS

AWS O Parallel Computing Service usa AWS Identity and Access Management funções vinculadas a serviços (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao AWS PCS. As funções vinculadas ao serviço são predefinidas pelo AWS PCS e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração do AWS PCS porque você não precisa adicionar manualmente as permissões necessárias. AWS O PCS define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, somente o AWS PCS pode assumir suas funções. As permissões definidas incluem as políticas de confiança e de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Você só pode excluir um perfil vinculado a serviço depois de excluir os recursos relacionados. Isso protege seus recursos do AWS PCS porque você não pode remover acidentalmente a permissão para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte AWS Serviços que funcionam com IAM e procure os serviços que têm Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Permissões de função vinculadas ao serviço para PCS AWS

AWS O PCS usa a função vinculada ao serviço chamada AWSServiceRoleForPCS — Concede permissão ao AWS PCS para gerenciar recursos do Amazon EC2.

A função vinculada ao serviço AWSService RoleFor PCS confia nos seguintes serviços para assumir a função:

- `pcs.amazonaws.com`

A política de permissões de função nomeada AWSPCSServiceRolePolicypermite que o AWS PCS conclua ações em recursos específicos.

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para obter mais informações, consulte Permissões do perfil vinculado a serviço no Guia do usuário do IAM.

Criação de uma função vinculada a serviços para PCS AWS

Você não precisa criar manualmente uma função vinculada ao serviço. AWS O PCS cria uma função vinculada ao serviço para você quando você cria um cluster.

Editando uma função vinculada ao serviço para PCS AWS

AWS O PCS não permite que você edite a função vinculada ao serviço AWSService RoleFor PCS. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias

entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para saber mais, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluindo uma função vinculada ao serviço para PCS AWS

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

> **Note**
>
> Se o serviço AWS PCS estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para remover recursos do AWS PCS usados pelo AWSService RoleFor PCS

Você deve excluir todos os seus clusters para excluir a função vinculada ao serviço AWSService RoleFor PCS. Para obter mais informações, consulte Excluir um cluster.

Como excluir manualmente o perfil vinculado ao serviço usando o IAM

Use o console do IAM AWS CLI, o ou a AWS API para excluir a função vinculada ao serviço AWSService RoleFor PCS. Para saber mais, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões suportadas para funções vinculadas ao serviço AWS PCS

AWS O PCS oferece suporte ao uso de funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Regiões e endpoints da AWS.

Função spot do Amazon EC2 para PCS AWS

Se você quiser criar um grupo de nós de computação AWS PCS que use o Spot como opção de compra, você também deve ter a função vinculada ao serviço AWSServiceRoleForEC2Spot em seu. Conta da AWS Você pode usar o AWS CLI comando a seguir para criar a função. Para obter mais informações, consulte Criar uma função vinculada ao serviço e Criar uma função para delegar permissões a um AWS serviço no Guia do AWS Identity and Access Management usuário.

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

Note

Você receberá o seguinte erro se Conta da AWS já tiver uma função `AWSServiceRoleForEC2Spot` do IAM.

```
 An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
operation: Service role name AWSServiceRoleForEC2Spot has been taken in this
account, please try a different suffix.
```

Permissões mínimas para AWS PCS

Esta seção descreve as permissões mínimas do IAM necessárias para que uma identidade do IAM (usuário, grupo ou função) use o serviço.

Sumário

Permissões mínimas para usar ações de API

Ação da API	Permissões mínimas	Permissões adicionais para o console
CreateCluster	`ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets,`	

Ação da API	Permissões mínimas	Permissões adicionais para o console
	`ec2:DescribeSecurityGroups`, `ec2:GetSecurityGroupsForVpc`, `iam:CreateServiceLinkedRole`, `secretsmanager:CreateSecret`, `secretsmanager:TagResource`, `secretsmanager:RotateSecret`, `pcs:CreateCluster`	
ListClusters	`pcs:ListClusters`	
GetCluster	`pcs:GetCluster`	`ec2:DescribeSubnets`
DeleteCluster	`pcs:DeleteCluster`	

Ação da API	Permissões mínimas	Permissões adicionais para o console
CreateComputeNodeGroup	`ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:DescribeInstanceTypeOfferings, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags, iam:PassRole, iam:GetInstanceProfile, pcs:CreateComputeNodeGroup`	`iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster`
ListComputerNodeGroups	`pcs:ListComputeNodeGroups`	`pcs:GetCluster`
GetComputeNodeGroup	`pcs:GetComputeNodeGroup`	`ec2:DescribeSubnets`

Ação da API	Permissões mínimas	Permissões adicionais para o console
UpdateComputeNodeGroup	`ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:DescribeInstanceTypeOfferings, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags, iam:PassRole, iam:GetInstanceProfile, pcs:UpdateComputeNodeGroup`	`pcs:GetComputeNodeGroup, iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster`
DeleteComputeNodeGroup	`pcs:DeleteComputeNodeGroup`	
CreateQueue	`pcs:CreateQueue`	`pcs:ListComputeNodeGroups, pcs:GetCluster`
ListQueues	`pcs:ListQueues`	`pcs:GetCluster`
GetQueue	`pcs:GetQueue`	

Ação da API	Permissões mínimas	Permissões adicionais para o console
UpdateQueue	`pcs:UpdateQueue`	`pcs:ListComputeNodeGroups, pcs:GetQueue`
DeleteQueue	`pcs:DeleteQueue`	

Permissões mínimas para usar tags

As permissões a seguir são necessárias para usar tags com seus recursos no AWS PCS.

```
pcs:ListTagsForResource,
pcs:TagResource,
pcs:UntagResource
```

Permissões mínimas para suportar registros

AWS O PCS envia dados de log para o Amazon CloudWatch Logs (CloudWatch Logs). Você deve garantir que sua identidade tenha as permissões mínimas para usar o CloudWatch Logs. Para obter mais informações, consulte Visão geral do gerenciamento de permissões de acesso aos seus recursos de CloudWatch registros no Guia do usuário do Amazon CloudWatch Logs.

Para obter informações sobre as permissões necessárias para que um serviço envie CloudWatch registros para o Logs, consulte Habilitar o registro de AWS serviços no Guia do usuário do Amazon CloudWatch Logs.

Permissões mínimas para usar blocos de capacidade

O Amazon EC2 Capacity Blocks for ML é uma opção de compra do Amazon EC2 que permite que você pague antecipadamente para reservar instâncias de computação acelerada baseadas em GPU dentro de um intervalo específico de data e hora para suportar cargas de trabalho de curta duração. Para obter mais informações, consulte Usando blocos de capacidade do Amazon EC2 para ML com PCS AWS.

Você escolhe usar blocos de capacidade ao criar ou atualizar um grupo de nós de computação. A identidade do IAM que você usa para criar ou atualizar o grupo de nós de computação precisa ter a seguinte permissão:

```
ec2:DescribeCapacityReservations
```

Permissões mínimas para um administrador de serviços

A política do IAM a seguir especifica as permissões mínimas necessárias para que uma identidade do IAM (usuário, grupo ou função) configure e gerencie o serviço AWS PCS.

> **Note**
>
> Os usuários que não configuram e gerenciam o serviço não precisam dessas permissões. Os usuários que executam apenas trabalhos usam o secure shell (SSH) para se conectar ao cluster. AWS Identity and Access Management (IAM) não lida com autenticação ou autorização para SSH.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PCSAccess",
      "Effect": "Allow",
      "Action": [
        "pcs:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2Access",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeImages",
        "ec2:GetSecurityGroupsForVpc",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:DescribeLaunchTemplates",
```

```
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:RunInstances",
        "ec2:CreateFleet",
        "ec2:CreateTags",
        "ec2:DescribeCapacityReservations"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamInstanceProfile",
      "Effect": "Allow",
      "Action": [
        "iam:GetInstanceProfile"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/*/AWSPCS*",
        "arn:aws:iam::*:role/AWSPCS*",
        "arn:aws:iam::*:role/aws-pcs/*",
        "arn:aws:iam::*:role/*/aws-pcs/*"
      ],
      "Condition": {
        "StringEquals": {
           "iam:PassedToService": [
             "ec2.amazonaws.com"
           ]
        }
      }
    },
    {
      "Sid": "SLRAccess",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
```

```
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleFor*",
        "arn:aws:iam::*:role/aws-service-role/spot.amazonaws.com/AWSServiceRoleFor*"
      ],
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": [
            "pcs.amazonaws.com",
            "spot.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AccessKMSKey",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SecretManagementAccess",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret",
        "secretsmanager:TagResource",
        "secretsmanager:UpdateSecret",
        "secretsmanager:RotateSecret"
      ],
      "Resource": "*"
    },
    {
       "Sid": "ServiceLogsDelivery",
       "Effect": "Allow",
       "Action": [
         "pcs:AllowVendedLogDeliveryForResource",
         "logs:PutDeliverySource",
         "logs:PutDeliveryDestination",
         "logs:CreateDelivery"
```

```
        ],
        "Resource": "*"
    }
  ]
}
```

Perfis de instância do IAM para o AWS Parallel Computing Service

Os aplicativos executados em uma instância do EC2 devem incluir AWS credenciais em todas as solicitações de AWS API que fizerem. Recomendamos que você use uma função do IAM para gerenciar credenciais temporárias na instância do EC2. Você pode definir um perfil de instância para fazer isso e anexá-lo às suas instâncias. Para obter mais informações, consulte as funções do IAM para o Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.

> **Note**
>
> Quando você usa o Console de gerenciamento da AWS para criar uma função do IAM para o Amazon EC2, o console cria um perfil de instância automaticamente e dá a ele o mesmo nome da função do IAM. Se você usa as AWS CLI ações de AWS API ou um AWS SDK para criar a função do IAM, você cria o perfil da instância como uma ação separada. Para obter mais informações, consulte Perfis de instância no Guia do usuário do Amazon Elastic Compute Cloud.

Você deve especificar o Amazon Resource Name (ARN) de um perfil de instância ao criar grupos de nós de computação. Você pode escolher perfis de instância diferentes para alguns ou todos os grupos de nós de computação.

Requisitos

Papel do IAM do perfil da instância

A função do IAM associada ao perfil da instância deve ter `/aws-pcs/` em seu caminho ou seu nome deve começar com`AWSPCS`.

Exemplo de função do IAM ARNs

- `arn:aws:iam::*:role/AWSPCS-example-role-1`
- `arn:aws:iam::*:role/aws-pcs/example-role-2`

Permissões

A função do IAM associada ao perfil da instância para AWS PCS deve incluir a política a seguir.

JSON

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

Políticas adicionais

Considere adicionar políticas gerenciadas ao perfil da instância. Por exemplo:

- O AmazonS3 ReadOnlyAccess fornece acesso somente de leitura a todos os buckets do S3.
- A Amazon SSMManaged InstanceCore habilita a funcionalidade principal do serviço AWS Systems Manager, como acesso remoto diretamente do Amazon Management Console.
- CloudWatchAgentServerPolicycontém as permissões necessárias para uso AmazonCloudWatchAgent em servidores.

Você também pode incluir suas próprias políticas de IAM que ofereçam suporte ao seu caso de uso específico.

Crie um perfil de instância para AWS PCS

AWS PCS console

Selecione Criar um perfil básico ao criar um grupo de nós de computação para que o AWS PCS crie um para você com a política mínima exigida.

Amazon EC2 console

Você pode criar um perfil de instância diretamente do console do Amazon EC2. Para obter mais informações, consulte Como usar perfis de instância no Guia AWS Identity and Access Management do usuário.

> **Important**
>
> Certifique-se de usar o prefixo necessário `AWSPCS` no nome da função do IAM.

AWS CLI

Configurando o perfil de instância básica usando o AWS CLI

> **Note**
>
> Substitua *`example-role`* nos exemplos a seguir pelo nome da sua função do IAM.

1. Crie uma função do IAM com `/aws-pcs/` o atributo de caminho ou um nome que comece com`AWSPCS`.

   a. Copie e cole o conteúdo a seguir em um novo arquivo de texto chamado`trust_policy.json`.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
```

```
    ]
}
```

b. Use um dos comandos a seguir para criar a função do IAM.

```
aws iam create-role --path /aws-pcs/ --role-name example-role --assume-role-
policy-document file://trust_policy.json
```

or

```
aws iam create-role --role-name AWSPCS-example-role --assume-role-policy-
document file://trust_policy.json
```

2. Anexe permissões.

a. Copie e cole o conteúdo a seguir em um novo arquivo de texto chamado`policy_document.json`.

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

b. Anexe o documento de política à função. Esse comando anexa a política como uma política embutida.

```
aws iam put-role-policy \
    --role-name example-role \
    --policy-name pcsRegisterInstancePolicy \
    --policy-document file://policy_document.json
```

3. Crie um perfil de instância. ***example-profile***Substitua pelo nome do seu perfil de instância.

```
aws iam create-instance-profile --instance-profile-name example-profile
```

4. Associe a função do IAM ao perfil da instância.

```
aws iam add-role-to-instance-profile \
   --instance-profile-name example-profile \
   --role-name example-role
```

Encontre perfis de instância usados com o AWS PCS

1. Se você não souber os nomes exatos das suas funções do IAM para AWS PCS, use o AWS CLI comando a seguir para listar as funções do IAM que atendem aos requisitos de nome do AWS PCS.

```
aws iam list-roles --query "Roles[?starts_with(RoleName, 'AWSPCS') ||
 contains(Path, '/aws-pcs/')].[RoleName]" --output text
```

2. Use o AWS CLI comando a seguir para listar os perfis de instância associados a uma função específica do IAM. *role-name*Substitua pelo nome de uma função do IAM que atenda aos requisitos de nome do AWS PCS.

```
aws iam list-instance-profiles-for-role --role-name role-name
```

Solução de problemas de identidade e acesso ao serviço de computação AWS paralela

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o AWS PCS e o IAM.

Tópicos

- Não estou autorizado a realizar uma ação no AWS PCS
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos do AWS PCS

Não estou autorizado a realizar uma ação no AWS PCS

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `pcs:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 pcs:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `pcs:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS PCS.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para realizar uma ação no AWS PCS. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos do AWS PCS

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o AWS PCS oferece suporte a esses recursos, consulteComo o serviço de computação AWS paralela funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Validação de conformidade para o serviço de computação AWS paralela

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis

e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a Documentação AWS de segurança.

Resiliência no serviço de computação AWS paralela

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte Infraestrutura AWS global.

Segurança de infraestrutura no serviço de computação AWS paralela

Como um serviço gerenciado, o AWS Parallel Computing Service é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o AWS PCS pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Quando o AWS PCS cria um cluster, o serviço inicia o controlador Slurm em uma conta de propriedade do serviço, separada dos nós de computação em sua conta. Para unir a comunicação entre o controlador e os nós de computação, o AWS PCS cria uma interface de rede elástica (ENI)

entre contas em sua VPC. O controlador Slurm usa o ENI para gerenciar e se comunicar com os nós de computação em diferentes Contas da AWS, mantendo a segurança e o isolamento dos recursos e, ao mesmo tempo, facilitando a eficiência da HPC e das operações. AI/ML

Análise e gerenciamento de vulnerabilidades no Serviço de Computação AWS Paralela

A configuração e os controles de TI são uma responsabilidade compartilhada entre você AWS e você. Para obter mais informações, consulte o modelo de responsabilidade AWS compartilhada. AWS lida com tarefas básicas de segurança para a infraestrutura subjacente na conta de serviço, como corrigir o sistema operacional nas instâncias do controlador, configuração do firewall e recuperação de desastres da AWS infraestrutura. Esses procedimentos foram revisados e certificados por terceiros certificados. Para obter mais detalhes, consulte Práticas recomendadas de segurança, identidade e conformidade.

> **Note**
>
> Os controladores Slurm não estão disponíveis enquanto os atualizamos. Os trabalhos em execução não são afetados. Os trabalhos enviados quando o controlador do cluster não está disponível são mantidos até que o controlador esteja disponível.

Você é responsável pela segurança da infraestrutura subjacente em seu Conta da AWS:

- Mantenha seu código, incluindo atualizações e patches de segurança.
- Corrija e atualize o sistema operacional na Amazon Machine Image (AMI) para seus grupos de nós de computação e atualize seus grupos de nós de computação para usar a AMI atualizada.
- Atualize o agendador para mantê-lo dentro das versões compatíveis. Atualize a AMI para seus grupos de nós de computação e atualize seu grupo de nós de computação para usar a AMI atualizada.
- Autentique e criptografe a comunicação entre os clientes do usuário e os nós aos quais eles se conectam.

Para obter mais informações sobre como atualizar a AMI para seus grupos de nós de computação, consulteAmazon Machine Images (AMIs) para AWS PCS.

Prevenção do problema "confused deputy" entre serviços

O problema de adjunto confuso é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir outra entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as `aws:SourceArn`chaves de contexto de condição `aws:SourceAccount`global nas políticas de recursos para limitar as permissões que o Serviço de Computação AWS Paralela (AWS PCS) concede a outro serviço ao recurso. Use `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curinga (*) para as partes desconhecidas do ARN. Por exemplo, .`arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`

Se o valor de `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deverá usar ambas as chaves de contexto de condição global para limitar as permissões.

O valor de `aws:SourceArn` deve ser um ARN de cluster.

O exemplo a seguir mostra como você pode usar as chaves de contexto de condição `aws:SourceAccount` global `aws:SourceArn` e as chaves de contexto no AWS PCS para evitar o confuso problema substituto.

```
{
"Version": "2012-10-17",
  "Statement": {
"Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
```

```
      "Service": "pcs.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": [
          "arn:aws:pcs:us-east-1:123456789012:cluster/*"
        ]
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Função do IAM para instâncias do Amazon EC2 provisionadas como parte de um grupo de nós de computação

AWS O PCS orquestra automaticamente a capacidade do Amazon EC2 para cada um dos grupos de nós de computação configurados em um cluster. Ao criar um grupo de nós de computação, os usuários devem fornecer um perfil de instância do IAM por meio do `iamInstanceProfileArn` campo. O perfil da instância especifica as permissões associadas às instâncias EC2 provisionadas. AWS O PCS aceita qualquer função que tenha `AWSPCS` como prefixo do nome da função ou `/aws-pcs/` como parte do caminho da função. A `iam:PassRole` permissão é necessária na identidade do IAM (usuário ou função) que cria ou atualiza um grupo de nós de computação. Quando um usuário chama as ações da `UpdateComputeNodeGroup` API `CreateComputeNodeGroup` ou da API, o AWS PCS verifica se o usuário tem permissão para realizar a `iam:PassRole` ação.

O exemplo de política a seguir concede permissões para passar somente perfis do IAM cujo nome comece com `AWSPCS`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/AWSPCS*",
            "Condition": {
                "StringEquals": {
```

```
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Melhores práticas de segurança para serviços de computação AWS paralela

Esta seção descreve as melhores práticas de segurança específicas do Serviço de Computação AWS Paralela (AWS PCS). Para saber mais sobre as melhores práticas de segurança em AWS, consulte Melhores práticas de segurança, identidade e conformidade.

Segurança relacionada à AMI

- Não use a amostra AWS PCS AMIs para cargas de trabalho de produção. A amostra não AMIs tem suporte e é destinada apenas para testes.
- Atualize regularmente o sistema operacional e o software na AMI para seus grupos de nós de computação para reduzir as vulnerabilidades.
- Use somente pacotes AWS PCS oficiais autenticados baixados de AWS fontes oficiais.
- Atualize regularmente os pacotes AWS PCS na AMI para grupos de nós de computação e atualize os nós de computação para usar a AMI atualizada. Considere automatizar esse processo para minimizar as vulnerabilidades.

Para obter mais informações, consulte Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

Segurança do Slurm Workload Manager

- Implemente controles de acesso e restrições de rede para proteger os nós de controle e computação do Slurm. Só permita que usuários e sistemas confiáveis enviem trabalhos e acessem os comandos de gerenciamento do Slurm.
- Use os recursos de segurança integrados do Slurm, como a autenticação do Slurm, para garantir que os envios de trabalhos e as comunicações sejam autenticados.

- Atualize as versões do Slurm para manter as operações e o suporte ao cluster sem problemas.

  > **Important**
  >
  > Qualquer cluster que usa uma versão do Slurm que tenha atingido o fim da vida útil do suporte (EOSL) é interrompido imediatamente. Use o link na parte superior das páginas do guia do usuário para assinar o feed RSS da documentação do AWS PCS e receber uma notificação quando uma versão do Slurm se aproximar do EOSL.

  Para obter mais informações, consulte Versões Slurm no PCS AWS.
- Alterne regularmente os segredos do cluster para manter a conformidade de segurança e corrigir possíveis comprometimentos de segurança. Isso é necessário para conformidade com HIPAA e FedRAMP.

  Para obter mais informações, consulte Segredos do cluster rotativo no AWS PCS.

Monitorar e registrar em log

- Use o Amazon CloudWatch Logs e AWS CloudTrail para monitorar e registrar ações em seus clusters Conta da AWS e. Use os dados para solução de problemas e auditoria.

Segurança de rede

- Implante seus clusters de AWS PCS em uma VPC separada para isolar seu ambiente de HPC de outros tráfegos de rede.
- Use grupos de segurança e listas de controle de acesso à rede (ACLs) para controlar o tráfego de entrada e saída para instâncias e sub-redes do AWS PCS.
- Use AWS PrivateLink nossos endpoints VPC para manter o tráfego de rede entre seus clusters e outros AWS serviços dentro da rede. AWS Para obter mais informações, consulte Acesso Serviço de Computação Paralela da AWS usando um endpoint de interface ()AWS PrivateLink.

Registro e monitoramento para AWS PCS

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do AWS PCS e de seus outros recursos da AWS. A AWS fornece as seguintes ferramentas de monitoramento para monitorar o AWS PCS, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos em que você executa AWS em tempo real. Você pode 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 CloudWatch rastrear o uso da CPU ou outras métricas de suas instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log a partir de instâncias do Amazon EC2 e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- AWS CloudTrailcaptura chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

Registros de conclusão de trabalhos no AWS PCS

Os registros de conclusão do trabalho fornecem detalhes importantes sobre seus trabalhos do Serviço de Computação AWS Paralela (AWS PCS) quando eles são concluídos, sem custo adicional. Você pode usar outros AWS serviços para acessar e processar seus dados de log, como Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3) e Amazon Data Firehose AWS ; o PCS registra metadados sobre seus trabalhos, como os seguintes.

- ID e nome do Job
- Informações do usuário e do grupo
- Estado do trabalho (como`COMPLETED`,`FAILED`,`CANCELLED`)

- Partição usada
- Limites de tempo
- Horários de início, término, envio e qualificáveis
- Lista e contagem de nós
- Contagem de processadores
- Diretório de trabalho
- Uso de recursos (CPU, memória)
- Códigos de saída
- Detalhes do nó (nomes, instância IDs, tipos de instância)

Sumário

Pré-requisitos

O diretor do IAM que gerencia o cluster AWS PCS deve permitir a `pcs:AllowVendedLogDeliveryForResource` ação.

O exemplo a seguir da política do IAM concede as permissões necessárias.

JSON

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

```
            "Effect": "Allow",
            "Action": ["pcs:AllowVendedLogDeliveryForResource"],
            "Resource": [
               "arn:aws:pcs:*::cluster/*"
            ]
         }
      ]
}
```

Configurar registros de conclusão do trabalho

Você pode configurar registros de conclusão de tarefas para seu cluster AWS PCS com o Console de gerenciamento da AWS ou AWS CLI.

Console de gerenciamento da AWS

Para configurar registros de conclusão de trabalhos com o console

1. Abra o console AWS PCS.
2. No painel de navegação, escolha Clusters.
3. Escolha o cluster ao qual você deseja adicionar os registros de conclusão do trabalho.
4. Na página de detalhes do cluster, escolha a guia Registros.
5. Em Job Conclution Logs, escolha Add para adicionar até 3 destinos de entrega de CloudWatch logs entre Logs, Amazon S3 e Firehose.
6. Escolha Atualizar entregas de registros.

AWS CLI

Para configurar registros de conclusão do trabalho com o AWS CLI

1. Crie um destino de entrega de registros:

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

Substitua:

- *region*— O Região da AWS local onde você deseja criar o destino, como `us-east-1`
- *pcs-logs-destination*— Um nome para o destino
- *resource-arn*— O Amazon Resource Name (ARN) de um grupo de CloudWatch logs do Logs, bucket do S3 ou stream de entrega do Firehose.

Para obter mais informações, consulte PutDeliveryDestinationa Referência da API Amazon CloudWatch Logs.

2. Defina o cluster PCS como uma fonte de entrega de registros:

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_JOBCOMP_LOGS
```

Substitua:

- *region*— O Região da AWS do seu cluster, como `us-east-1`
- *cluster-logs-source-name*— Um nome para a fonte
- *cluster-arn*— o ARN do seu AWS cluster PCS

Para obter mais informações, consulte PutDeliverySourcea Referência da API Amazon CloudWatch Logs.

3. Conecte a fonte de entrega ao destino da entrega:

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

Substitua:

- *region*— O Região da AWS, como `us-east-1`
- *cluster-logs-source*— O nome da sua fonte de entrega
- *destination-arn*— O ARN do seu destino de entrega

Para obter mais informações, consulte CreateDeliverya Referência da API Amazon CloudWatch Logs.

Como encontrar registros de conclusão de trabalhos

Você pode configurar destinos de log no CloudWatch Logs e no Amazon S3. AWS O PCS usa os seguintes nomes de caminhos estruturados e nomes de arquivos.

CloudWatch Registros

AWS O PCS usa o seguinte formato de nome para o stream de CloudWatch registros:

```
AWSLogs/PCS/cluster-id/jobcomp.log
```

Por exemplo: `AWSLogs/PCS/pcs_abc123de45/jobcomp.log`

Amazon S3

AWS O PCS usa o seguinte formato de nome para o caminho do S3:

```
AWSLogs/account-id/PCS/region/cluster-id/jobcomp/year/month/day/hour/
```

Por exemplo: `AWSLogs/111122223333/PCS/us-east-1/pcs_abc123de45/jobcomp/2025/06/19/11/`

AWS O PCS usa o seguinte formato de nome para os arquivos de log:

```
PCS_jobcomp_year-month-day-hour_cluster-id_random-id.log.gz
```

Por exemplo: `PCS_jobcomp_2025-06-19-11_pcs_abc123de45_04be080b.log.gz`

Campos do registro de conclusão do trabalho

AWS O PCS grava dados de registro de conclusão do trabalho como objetos JSON. O contêiner JSON `jobcomp` contém os detalhes do trabalho. A tabela a seguir descreve os campos dentro do `jobcomp` contêiner. Alguns campos só estão presentes em circunstâncias específicas, como para trabalhos de matriz ou trabalhos heterogêneos.

Campos do registro de conclusão do trabalho

Nome	Valor de exemplo	Obrigatório	Observações
`job_id`	`11`	sim	Sempre presente com valor
`user`	`"root"`	sim	Sempre presente com valor
`user_id`	`0`	sim	Sempre presente com valor
`group`	`"root"`	sim	Sempre presente com valor
`group_id`	`0`	sim	Sempre presente com valor
`name`	`"wrap"`	sim	Sempre presente com valor
`job_state`	`"COMPLETED"`	sim	Sempre presente com valor
`partition`	`"Hydra-MpiQueue-abcdef01-7"`	sim	Sempre presente com valor
`time_limit`	`"UNLIMITED"`	sim	Sempre presente, mas pode estar `"UNLIMITED"`
`start_time`	`"2025-06-19T10:58:57"`	sim	Sempre presente, mas pode estar `"Unknown"`
`end_time`	`"2025-06-19T10:58:57"`	sim	Sempre presente, mas pode estar `"Unknown"`
`node_list`	`"Hydra-MpiNG-abcdef01-2345-1"`	sim	Sempre presente com valor
`node_cnt`	`1`	sim	Sempre presente com valor

Nome	Valor de exemplo	Obrigatório	Observações
`proc_cnt`	`1`	sim	Sempre presente com valor
`work_dir`	`"/root"`	sim	Sempre presente, mas pode estar `"Unknown"`
`reservation_name`	`"weekly_maintenance"`	sim	Sempre presente, mas pode ser uma string vazia `""`
`tres.cpu`	`1`	sim	Sempre presente com valor
`tres.mem.val`	`600`	sim	Sempre presente com valor
`tres.mem.unit`	`"M"`	sim	Pode ser `"M"` ou `"bb"`
`tres.node`	`1`	sim	Sempre presente com valor
`tres.billing`	`1`	sim	Sempre presente com valor
`account`	`"finance"`	sim	Sempre presente, mas pode ser uma string vazia `""`
`qos`	`"normal"`	sim	Sempre presente, mas pode ser uma string vazia `""`
`wc_key`	`"project_1"`	sim	Sempre presente, mas pode ser uma string vazia `""`
`cluster`	`"unknown"`	sim	Sempre presente, mas pode estar `"unknown"`
`submit_time`	`"2025-06-19T10:55:46"`	sim	Sempre presente, mas pode estar `"Unknown"`

Nome	Valor de exemplo	Obrigatório	Observações
`eligible_time`	`"2025-06-19T10:55:46"`	sim	Sempre presente, mas pode estar `"Unknown"`
`array_job_id`	`12`	não	Presente somente se o trabalho for um trabalho de matriz
`array_task_id`	`1`	não	Presente somente se o trabalho for um trabalho de matriz
`het_job_id`	`10`	não	Presente apenas se o trabalho for heterogêneo
`het_job_offset`	`0`	não	Presente apenas se o trabalho for heterogêneo
`derived_exit_code_status`	`0`	sim	Sempre presente com valor
`derived_exit_code_signal`	`0`	sim	Sempre presente com valor
`exit_code_status`	`0`	sim	Sempre presente com valor
`exit_code_signal`	`0`	sim	Sempre presente com valor
`node_details[0].name`	`"Hydra-MpiNG-abcdef01-2345-1"`	não	Sempre presente, mas `node_details` pode estar `"[]"`

Nome	Valor de exemplo	Obrigatório	Observações
`node_details[0].instance_id`	`"i-0abcdef01234567a"`	não	Sempre presente, mas `node_details` pode estar `"[]"`
`node_details[0].instance_type`	`"t4g.micro"`	não	Sempre presente, mas `node_details` pode estar `"[]"`

Exemplos de registros de conclusão de trabalhos

Os exemplos a seguir mostram registros de conclusão de trabalhos para vários tipos e estados de trabalhos:


```
{ "jobcomp": { "job_id": 1, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:32:57",
 "end_time": "2025-06-19T16:33:03", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name":
 "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
 "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
 "submit_time": "2025-06-19T16:29:40", "eligible_time": "2025-06-19T16:29:41",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name":
 "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type":
 "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 2, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:33:13",
 "end_time": "2025-06-19T16:33:14", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name":
 "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
 "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
 "submit_time": "2025-06-19T16:33:13", "eligible_time": "2025-06-19T16:33:13",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
```

```
 "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name":
 "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type":
 "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 3, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T22:58:57", "end_time":
 "2025-06-19T22:58:57", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T22:55:46",
 "eligible_time": "2025-06-19T22:55:46", "derived_exit_code_status": 0,
 "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal":
 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id":
 "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 4, "user": "root", "user_id": 0, "group": "root",
 "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-
MpiQueue-abcdef01-7", "time_limit": "525600", "start_time": "2025-06-19T23:04:27",
 "end_time": "2025-06-19T23:04:27", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/root", "reservation_name":
 "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
 "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
 "submit_time": "2025-06-19T23:01:38", "eligible_time": "2025-06-19T23:01:38",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" }, { "name":
 "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def345abc67890", "instance_type":
 "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 5, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "FAILED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:00", "end_time":
 "2025-06-19T23:09:00", "node_list": "(null)", "node_cnt": 0, "proc_cnt": 0,
 "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val":
 1, "unit": "G" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key":
 "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:00", "eligible_time":
 "2025-06-19T23:09:00", "derived_exit_code_status": 0, "derived_exit_code_signal": 0,
 "exit_code_status": 0, "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 6, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:36",
 "end_time": "2025-06-19T23:09:36", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
 { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
 "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:35",
 "eligible_time": "2025-06-19T23:09:36", "het_job_id": 6, "het_job_offset": 0,
```

```
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 7, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:10:03",
 "end_time": "2025-06-19T23:10:03", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
 { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
 "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:10:03",
 "eligible_time": "2025-06-19T23:10:03", "het_job_id": 7, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 8, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
 "2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
 "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 9, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
 "2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
 "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
 "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 10, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
 "2025-06-19T23:12:24", "node_list":"Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
 "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
```

```
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 11, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
 "2025-06-19T23:12:24", "node_list":"Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 600, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
 "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
 "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 13, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:57", "end_time":
 "2025-06-19T23:47:58", "node_list":"Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
 "eligible_time": "2025-06-19T23:43:56" , "array_job_id": 12, "array_task_id": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 12, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:58", "end_time":
 "2025-06-19T23:47:58", "node_list":"Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
 "eligible_time": "2025-06-19T23:43:56" , "array_job_id": 12, "array_task_id": 2,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
```


Logs do agendador no AWS PCS

Você pode configurar o AWS PCS para enviar dados de registro detalhados do seu agendador de cluster para o Amazon CloudWatch Logs, o Amazon Simple Storage Service (Amazon S3) e o Amazon Data Firehose. Isso pode ajudar no monitoramento e na solução de problemas.

Sumário

Pré-requisitos

O diretor do IAM que gerencia o cluster AWS PCS deve permitir a `pcs:AllowVendedLogDeliveryForResource` ação.

O exemplo a seguir da política do IAM concede as permissões necessárias.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "PcsAllowVendedLogsDelivery",
         "Effect": "Allow",
         "Action": ["pcs:AllowVendedLogDeliveryForResource"],
         "Resource": [
            "arn:aws:pcs:*::cluster/*"
         ]
      }
   ]
}
```

Configurar registros do agendador

Você pode configurar os registros do agendador para seu cluster AWS PCS com o Console de gerenciamento da AWS ou AWS CLI.

Console de gerenciamento da AWS

Para configurar os registros do agendador com o console

1. Abra o console AWS PCS.
2. No painel de navegação, escolha Clusters.
3. Escolha o cluster ao qual você deseja adicionar os registros do agendador.
4. Na página de detalhes do cluster, escolha a guia Registros.
5. Em Scheduler Logs, escolha Add para adicionar até 3 destinos de entrega de CloudWatch logs entre Logs, Amazon S3 e Firehose.
6. Escolha Atualizar entregas de registros.

AWS CLI

Para configurar os registros do agendador com o AWS CLI

1. Crie um destino de entrega de registros:

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

Substitua:

- *region*— O Região da AWS local onde você deseja criar o destino, como `us-east-1`
- *pcs-logs-destination*— Um nome para o destino
- *resource-arn*— O Amazon Resource Name (ARN) de um grupo de CloudWatch logs do Logs, bucket do S3 ou stream de entrega do Firehose.

Para obter mais informações, consulte PutDeliveryDestinationa Referência da API Amazon CloudWatch Logs.

2. Defina o cluster PCS como uma fonte de entrega de registros:

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
```

```
  --log-type PCS_SCHEDULER_LOGS
```

Substitua:

- *region*— O Região da AWS do seu cluster, como `us-east-1`
- *cluster-logs-source-name*— Um nome para a fonte
- *cluster-arn*— o ARN do seu AWS cluster PCS

Para obter mais informações, consulte PutDeliverySourcea Referência da API Amazon CloudWatch Logs.

3. Conecte a fonte de entrega ao destino da entrega:

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

Substitua:

- *region*— O Região da AWS, como `us-east-1`
- *cluster-logs-source*— O nome da sua fonte de entrega
- *destination-arn*— O ARN do seu destino de entrega

Para obter mais informações, consulte CreateDeliverya Referência da API Amazon CloudWatch Logs.

Caminhos e nomes do fluxo de registros do agendador

O caminho e o nome dos registros do agendador AWS PCS dependem do tipo de destino.

- CloudWatch Logs
  - Um stream de CloudWatch registros segue essa convenção de nomenclatura.

```
AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
```

Example

```
AWSLogs/PCS/abcdef0123/slurmctld_24.05.log
```

- Bucket do S3
  - Um caminho de saída do bucket S3 segue esta convenção de nomenclatura:

```
AWSLogs/${account-id}/PCS/${region}/${cluster_id}/${log_name}/
${scheduler_major_version}/yyyy/MM/dd/HH/
```

Example

```
AWSLogs/111111111111/PCS/us-east-2/abcdef0123/slurmctld/24.05/2024/09/01/00.
```

  - Um nome de objeto S3 segue esta convenção:

```
PCS_${log_name}_${scheduler_major_version}_#{expr date 'event_timestamp', format:
 "yyyy-MM-dd-HH"}_${cluster_id}_${hash}.log
```

Example

```
PCS_slurmctld_24.05_2024-09-01-00_abcdef0123_0123abcdef.log
```

Exemplo de registro de log do agendador

AWS Os registros do agendador PCS são estruturados. Eles incluem campos como identificador do cluster, tipo de agendador, versões principais e de patch, além da mensagem de log emitida pelo processo do controlador Slurm. Aqui está um exemplo.

```
{
    "resource_id": "s3431v9rx2",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1721230979,
    "log_level": "info",
    "log_name": "slurmctld",
    "scheduler_type": "slurm",
    "scheduler_major_version": "25.05",
    "scheduler_patch_version": "3",
```

```
    "node_type": "controller_primary",
    "message": "[2024-07-17T15:42:58.614+00:00] Running as primary controller\n"
}
```

Serviço de monitoramento de computação AWS paralela com a Amazon CloudWatch

CloudWatch A Amazon fornece monitoramento da integridade e do desempenho do seu cluster do AWS Parallel Computing Service (AWS PCS) coletando métricas do cluster em intervalos. Essas métricas são mantidas, permitindo que você acesse dados históricos e obtenha insights sobre o desempenho do seu cluster ao longo do tempo.

CloudWatch também permite monitorar as instâncias do EC2 lançadas pelo AWS PCS para atender aos seus requisitos de escalabilidade. Embora você possa inspecionar registros em instâncias em execução, CloudWatch as métricas e os dados de registro geralmente são excluídos quando as instâncias são encerradas. No entanto, você pode configurar o CloudWatch agente em instâncias usando um modelo de execução do EC2 para manter métricas e registros mesmo após o encerramento da instância, permitindo monitoramento e análise de longo prazo.

Explore os tópicos desta seção para saber mais sobre como monitorar o uso do AWS PCS CloudWatch.

Tópicos

- Monitorando métricas do AWS PCS usando CloudWatch
- Monitoramento de instâncias de AWS PCS usando a Amazon CloudWatch

Monitorando métricas do AWS PCS usando CloudWatch

Você pode monitorar a integridade do cluster AWS PCS usando a Amazon CloudWatch, que coleta dados do seu cluster e os transforma em métricas quase em tempo real. Essas estatísticas são mantidas por um período de 15 meses, para que você possa acessar informações históricas e ter uma perspectiva melhor sobre o desempenho do seu cluster. As métricas do cluster são enviadas CloudWatch em períodos de 1 minuto. Para obter mais informações sobre CloudWatch, consulte O que é a Amazon CloudWatch? no Guia do CloudWatch usuário da Amazon.

AWS O PCS publica as seguintes métricas no namespace AWS/PCS em. CloudWatch Eles têm uma única dimensão,`ClusterId`.

Name (Nome)	Description	Unidades
ActualCapacity	IdleCapacity + UtilizedCapacity	Contagem
CapacityUtilization	UtilizedCapacity / ActualCapacity	Contagem
DesiredCapacity	ActualCapacity + PendingCapacity	Contagem
IdleCapacity	Contagem de instâncias em execução, mas não alocadas para trabalhos	Contagem
UtilizedCapacity	Contagem de instâncias em execução e alocadas para trabalhos	Contagem

Monitoramento de instâncias de AWS PCS usando a Amazon CloudWatch

O AWS PCS lança instâncias do Amazon EC2 conforme necessário para atender aos requisitos de escalabilidade definidos em seus grupos de nós de computação do PCS. Você pode monitorar essas instâncias enquanto elas estão em execução usando a Amazon CloudWatch. Você pode inspecionar os registros das instâncias em execução fazendo login nelas e usando ferramentas de linha de comando interativas. No entanto, por padrão, os dados de CloudWatch métricas só são retidos por um período limitado quando uma instância é encerrada, e os registros da instância geralmente são excluídos junto com os volumes do EBS que apoiam a instância. Para reter métricas ou dados de registro das instâncias lançadas pelo PCS após o encerramento, você pode configurar o CloudWatch agente em suas instâncias com um modelo de execução do EC2. Este tópico fornece uma visão geral do monitoramento de instâncias em execução e fornece exemplos de como configurar métricas e registros de instâncias persistentes.

Monitorando instâncias em execução

Encontrando instâncias do AWS PCS

Para monitorar instâncias lançadas pelo PCS, encontre as instâncias em execução associadas a um cluster ou grupo de nós de computação. Em seguida, no console do EC2 de uma determinada instância, inspecione as seções Status e alarmes e Monitoramento. Se o acesso de login estiver configurado para essas instâncias, você poderá se conectar a elas e inspecionar vários arquivos de log nas instâncias. Para obter mais informações sobre como identificar quais instâncias são gerenciadas pelo PCS, consulteEncontrando instâncias de grupos de nós de computação no AWS PCS.

Habilitando métricas detalhadas

Por padrão, as métricas da instância são coletadas em intervalos de 5 minutos. Para coletar métricas em intervalos de um minuto, ative o CloudWatch monitoramento detalhado em seu modelo de lançamento do grupo de nós de computação. Para obter mais informações, consulte Ativar o CloudWatch monitoramento detalhado.

Configurando métricas e registros de instâncias persistentes

Você pode reter as métricas e os registros de suas instâncias instalando e configurando o CloudWatch agente da Amazon nelas. Isso consiste em três etapas principais:

1. Crie uma configuração de CloudWatch agente.
2. Armazene a configuração onde ela possa ser recuperada pelas instâncias do PCS.
3. Escreva um modelo de execução do EC2 que instale o software do CloudWatch agente, busque sua configuração e inicie o CloudWatch agente usando a configuração.

Para obter mais informações, consulte Coletar métricas, registros e rastreamentos com o CloudWatch agente no Guia CloudWatch do usuário da Amazon Usando modelos de lançamento do Amazon EC2 com PCS AWS e.

Criar uma configuração de CloudWatch agente

Antes de implantar o CloudWatch agente em suas instâncias, você deve gerar um arquivo de configuração JSON que especifique as métricas, os registros e os rastreamentos a serem coletados. Os arquivos de configuração podem ser criados usando um assistente ou manualmente, usando um editor de texto. O arquivo de configuração será criado manualmente para esta demonstração.

Em um computador em que você tenha a AWS CLI instalada, crie um arquivo de CloudWatch configuração chamado config.json com o conteúdo a seguir. Você também pode usar o seguinte URL para baixar uma cópia do arquivo.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

Observações

- Os caminhos de log no arquivo de amostra são para o Amazon Linux 2. Se suas instâncias usarem um sistema operacional básico diferente, altere os caminhos conforme apropriado.
- Para capturar outros registros, adicione outras entradas abaixo`collect_list`.
- Os valores em `{brackets}` são variáveis modeladas. Para obter a lista completa das variáveis suportadas, consulte Criar ou editar manualmente o arquivo de configuração do CloudWatch agente no Guia CloudWatch do usuário da Amazon.
- Você pode optar por omitir `logs` ou `metrics` se não quiser coletar esses tipos de informações.

```
{
    "agent": {
        "metrics_collection_interval": 60
    },
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/var/log/cloud-init.log",
                        "log_group_class": "STANDARD",
                        "log_group_name": "/PCSLogs/instances",
                        "log_stream_name": "{instance_id}.cloud-init.log",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/cloud-init-output.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.cloud-init-output.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/amazon/pcs/bootstrap.log",
```

```
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.bootstrap.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/slurmd.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.slurmd.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/messages",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.messages",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/secure",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.secure",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    }
                ]
            }
        }
    },
    "metrics": {
        "aggregation_dimensions": [
            [
                "InstanceId"
            ]
        ],
        "append_dimensions": {
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
            "ImageId": "${aws:ImageId}",
            "InstanceId": "${aws:InstanceId}",
            "InstanceType": "${aws:InstanceType}"
        },
        "metrics_collected": {
            "cpu": {
```

```
                "measurement": [
                    "cpu_usage_idle",
                    "cpu_usage_iowait",
                    "cpu_usage_user",
                    "cpu_usage_system"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ],
                "totalcpu": false
            },
            "disk": {
                "measurement": [
                    "used_percent",
                    "inodes_free"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "diskio": {
                "measurement": [
                    "io_time"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "mem": {
                "measurement": [
                    "mem_used_percent"
                ],
                "metrics_collection_interval": 60
            },
            "swap": {
                "measurement": [
                    "swap_used_percent"
                ],
                "metrics_collection_interval": 60
            }
        }
```

```
    }
}
```

Esse arquivo instrui o CloudWatch agente a monitorar vários arquivos que podem ser úteis no diagnóstico de erros na inicialização, autenticação e login da instância e em outros domínios de solução de problemas. Isso inclui:

- `/var/log/cloud-init.log`— Saída do estágio inicial da configuração da instância
- `/var/log/cloud-init-output.log`— Saída de comandos que são executados durante a configuração da instância
- `/var/log/amazon/pcs/bootstrap.log`— Saída de operações específicas do PC que são executadas durante a configuração da instância
- `/var/log/slurmd.log`— Saída do daemon slurmd do gerenciador de carga de trabalho Slurm
- `/var/log/messages`— Mensagens do sistema do kernel, serviços do sistema e aplicativos
- `/var/log/secure`— Registros relacionados a tentativas de autenticação, como SSH, sudo e outros eventos de segurança

Os arquivos de log são enviados para um grupo de CloudWatch log chamado`/PCSLogs/instances`. Os fluxos de log são uma combinação do ID da instância e do nome base do arquivo de log. O grupo de registros tem um tempo de retenção de 30 dias.

Além disso, o arquivo instrui o CloudWatch agente a coletar várias métricas comuns, agregando-as por ID da instância.

Armazene a configuração

O arquivo de configuração do CloudWatch agente precisa ser armazenado onde possa ser acessado pelas instâncias do nó de computação do PCS. Há duas maneiras comuns de fazer isso. Você pode carregá-lo em um bucket do Amazon S3 ao qual suas instâncias do grupo de nós computacionais terão acesso por meio de seu perfil de instância. Como alternativa, você pode armazená-lo como um parâmetro SSM no Amazon Systems Manager Parameter Store.

Fazer upload para um bucket do S3

Para armazenar seu arquivo no S3, use os comandos da AWS CLI a seguir. Antes de executar o comando, faça estas substituições:

- *`amzn-s3-demo-bucket`*Substitua pelo seu próprio nome de bucket do S3

Primeiro, (isso é opcional se você tiver um bucket existente), crie um bucket para armazenar seus arquivos de configuração.

```
aws s3 mb s3://amzn-s3-demo-bucket
```

Em seguida, faça o upload do arquivo para o bucket.

```
aws s3 cp ./config.json s3://amzn-s3-demo-bucket/
```

Armazenar como um parâmetro SSM

Para armazenar seu arquivo como um parâmetro SSM, use o comando a seguir. Antes de executar o comando, faça estas substituições:

- *region-code*Substitua pela região da AWS em que você está trabalhando com o AWS PCS.
- (Opcional) *AmazonCloudWatch-PCS* Substitua o parâmetro pelo seu próprio nome. Observe que, se você alterar o prefixo do nome de, `AmazonCloudWatch-` precisará adicionar especificamente o acesso de leitura ao parâmetro SSM no perfil da instância do seu grupo de nós.

```
aws ssm put-parameter \
   --region region-code \
   --name "AmazonCloudWatch-PCS" \
   --type String \
   --value file://config.json
```

Escreva um modelo de lançamento do EC2

Os detalhes específicos do modelo de lançamento dependem de seu arquivo de configuração estar armazenado no S3 ou no SSM.

Use uma configuração armazenada no S3

Esse script instala o CloudWatch agente, importa um arquivo de configuração de um bucket do S3 e inicia o CloudWatch agente com ele. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- *amzn-s3-demo-bucket*— O nome de um bucket do S3 que sua conta pode ler
- */config.json*— Caminho relativo à raiz do bucket do S3 em que a configuração está armazenada

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-cloudwatch-agent

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/config.json /etc/s3-cw-config.json
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
 ec2 -s -c file://etc/s3-cw-config.json

--==MYBOUNDARY==--
```

O perfil da instância do IAM para o grupo de nós deve ter acesso ao bucket. Aqui está um exemplo de política do IAM para o bucket no script de dados do usuário acima.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Observe também que as instâncias devem permitir tráfego de saída para o S3 e CloudWatch os endpoints. Isso pode ser feito usando grupos de segurança ou VPC endpoints, dependendo da arquitetura do cluster.

Use uma configuração armazenada no SSM

Esse script instala o CloudWatch agente, importa um arquivo de configuração de um parâmetro SSM e inicia o CloudWatch agente com ele. Substitua os seguintes valores nesse script pelos seus próprios detalhes:

- (Opcional) *AmazonCloudWatch-PCS* Substitua o parâmetro pelo seu próprio nome.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-cloudwatch-agent

runcmd:
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
 ec2 -s -c ssm:AmazonCloudWatch-PCS

--==MYBOUNDARY==--
```

A política de instância do IAM para o grupo de nós deve ter o CloudWatchAgentServerPolicyanexado a ela.

Se o nome do seu parâmetro não começar com, `AmazonCloudWatch-` você precisará adicionar especificamente o acesso de leitura ao parâmetro SSM em seu perfil de instância de grupo de nós. Aqui está um exemplo de política do IAM que ilustra isso para prefixo*DOC-EXAMPLE-PREFIX*.

JSON

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

```
      "Sid" : "CustomCwSsmMParamReadOnly",
      "Effect" : "Allow",
      "Action" : [
        "ssm:GetParameter"
      ],
      "Resource" : "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

Observe também que as instâncias devem permitir tráfego de saída para o SSM e CloudWatch os endpoints. Isso pode ser feito usando grupos de segurança ou VPC endpoints, dependendo da arquitetura do cluster.

Registrando chamadas de API do serviço de computação AWS paralela usando AWS CloudTrail

AWS O PCS é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no AWS PCS. CloudTrail captura todas as chamadas de API para AWS PCS como eventos. As chamadas capturadas incluem chamadas do console do AWS PCS e chamadas de código para as operações da API do AWS PCS. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para AWS PCS. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao AWS PCS, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

AWS Informações do PCS em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre no AWS PCS, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes no seu Conta da AWS. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em seu Conta da AWS, incluindo eventos para AWS PCS, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

Todas as ações do AWS PCS são registradas CloudTrail e documentadas na Referência da API do Serviço de Computação AWS Paralela. Por exemplo, chamadas para as `DeleteCluster` ações `CreateComputeNodeGroupUpdateQueue`, e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Compreendendo as entradas do arquivo de CloudTrail log do AWS PCS

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma `CreateQueue` ação.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
        "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
        "accountId": "012345678910",
        "accessKeyId": "ASIAY36PTPIEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAY36PTPIEEXAMPLE",
                "arn": "arn:aws:iam::012345678910:role/Admin",
                "accountId": "012345678910",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2024-07-16T17:05:51Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-07-16T17:13:09Z",
    "eventSource": "pcs.amazonaws.com",
    "eventName": "CreateQueue",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36",
    "requestParameters": {
        "clientToken": "c13b7baf-2894-42e8-acec-example",
        "clusterIdentifier": "abcdef0123",
        "computeNodeGroupConfigurations": [
            {
                "computeNodeGroupId": "abcdef0123"
            }
        ],
        "queueName": "all"
    },
    "responseElements": {
        "queue": {
```

```
            "arn": "arn:aws:pcs:us-east-1:609783872011:cluster/abcdef0123/queue/
abcdef0123",
            "clusterId": "abcdef0123",
            "computeNodeGroupConfigurations": [
                {
                    "computeNodeGroupId": "abcdef0123"
                }
            ],
            "createdAt": "2024-07-16T17:13:09.276069393Z",
            "id": "abcdef0123",
            "modifiedAt": "2024-07-16T17:13:09.276069393Z",
            "name": "all",
            "status": "CREATING"
        }
    },
    "requestID": "a9df46d7-3f6d-43a0-9e3f-example",
    "eventID": "7ab18f88-0040-47f5-8388-example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "012345678910",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "pcs.us-east-1.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

Endpoints e cotas de serviço para PCS AWS

As seções a seguir descrevem os endpoints e as cotas de serviço do Serviço de Computação AWS Paralela (AWS PCS). As cotas de serviço, anteriormente chamadas de limites, são o número máximo de recursos ou operações de serviço para você. Conta da AWS

Você Conta da AWS tem cotas padrão para cada AWS serviço. A menos que especificado de outra forma, cada cota é específica da região . Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Para obter mais informações, consulte Service Quotas da AWS, na Referência geral da AWS .

Sumário

Service endpoints

Nome da região	Região	Endpoint	Protocolo
Leste dos EUA (Ohio)	us-east-2	pcs.us-east-2.amazonaws.com pcs-fips.us-east-2.amazonaws.com pcs-fips.us-east-2.api.aws pcs.us-east-2.api.aws	HTTPS
Leste dos EUA (Norte da Virgínia)	us-east-1	pcs.us-east-1.amazonaws.com	HTTPS

Nome da região	Região	Endpoint	Protocolo
		pcs-fips.us-east-1.amazonaws.com pcs-fips.us-east-1.api.aws pcs.us-east-1.api.aws	
Oeste dos EUA (Oregon)	us-west-2	pcs.us-west-2.amazonaws.com pcs-fips.us-west-2.amazonaws.com pcs-fips.us-west-2.api.aws pcs.us-west-2.api.aws	HTTPS
Ásia-Pacífico (Singapura)	ap-southeast-1	pcs.ap-southeast-1.amazonaws.com pcs.ap-southeast-1.api.aws	HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	pcs.ap-southeast-2.amazonaws.com pcs.ap-southeast-2.api.aws	HTTPS
Ásia-Pacífico (Tóquio)	ap-northeast-1	pcs.ap-northeast-1.amazonaws.com pcs.ap-northeast-1.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
Europa (Frankfurt)	eu-central-1	pcs.eu-central-1.amazonaws.com pcs.eu-central-1.api.aws	HTTPS
Europa (Irlanda)	eu-west-1	pcs.eu-west-1.amazonaws.com pcs.eu-west-1.api.aws	HTTPS
Europa (Londres)	eu-west-2	pcs.eu-west-2.amazonaws.com pcs.eu-west-2.api.aws	HTTPS
Europa (Estocolmo)	eu-north-1	pcs.eu-north-1.amazonaws.com pcs.eu-north-1.api.aws	HTTPS
AWS GovCloud (Leste dos EUA)	us-gov-east-1	peças. us-gov-east-1.amazonaws.com dicas de peças. us-gov-east-1.amazonaws.com dicas de peças. us-gov-east-1.api.aws peças. us-gov-east-1.api.aws	HTTPS

Nome da região	Região	Endpoint	Protocolo
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	peças. us-gov-west-1.amazonaws.com dicas de peças. us-gov-west-1.amazonaws.com dicas de peças. us-gov-west-1.api.aws peças. us-gov-west-1.api.aws	HTTPS

Cotas de serviço

Nome	Padrão	Ajustável	Descrição
Clusters	5	Sim	O número máximo de clusters por Região da AWS.

> **Note**
>
> Os valores padrão são as cotas iniciais definidas por AWS. Esses valores padrão são separados do valor real da cota aplicada e das cotas de serviço máximas possíveis. Para obter mais informações, consulte Terminologia do Service Quotas no Guia do usuário do Service Quotas.

Essas cotas de serviço estão listadas em Serviço de Computação AWS Paralela (PCS) no Console de gerenciamento da AWS. Para solicitar um aumento de cota para valores que são mostrados como ajustáveis, consulte Solicitando um aumento de cota no Guia do usuário de Cotas de Serviço.

> **Important**
>
> Lembre-se de verificar a Região da AWS configuração atual no Console de gerenciamento da AWS.

Cotas internas

As cotas a seguir são internas e não são ajustáveis.

Nome	Padrão	Ajustável	Descrição
Criação simultânea de clusters	1	Não	Número máximo de clusters no estado `Creating` por Região da AWS.
Grupos de nós de computação por cluster	10	Não	O número máximo de grupos de nós de computação por cluster.
Filas por cluster	10	Não	O número máximo de filas por cluster.

Cotas relevantes para outros serviços AWS

AWS O PCS usa outros AWS serviços. Suas cotas de serviço para esses serviços afetam seu uso do AWS PCS.

Cotas de serviços do Amazon EC2 que afetam o PCS AWS

- Solicitações de instâncias spot
- Execução de instâncias sob demanda
- Modelos de inicialização
- Versões do modelo de execução
- Solicitações de API do Amazon EC2

Para obter mais informações, consulte as cotas de serviço do Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.

Solução de problemas no serviço de computação AWS paralela

Os tópicos a seguir fornecem orientação para solucionar alguns problemas que você pode encontrar no AWS PCS.

- Atualizações do cluster
- Problemas de bootstrap do nó de computação
- Configurações personalizadas do Slurm
- Instâncias do EC2 encerradas após a reinicialização
- Identidade e acesso
- Problemas de reinicialização do Slurm

Uma instância do EC2 no AWS PCS é encerrada e substituída após a reinicialização

Visão geral do problema

Depois que uma instância do EC2 em um grupo de nós de computação é reinicializada, o AWS PCS encerra e substitui automaticamente a instância.

Por que isso acontece

AWS O PCS não suporta reinicializações de instâncias. Se uma instância do EC2 for reinicializada, o AWS PCS considerará a instância não íntegra e a substituirá. Se o AWS PCS encerra e substitui continuamente suas instâncias, pode ser porque algo reinicializa suas instâncias após a inicialização. Alguns exemplos incluem reinicializações por automação na instância do EC2 (como uma reinicialização automática após a aplicação de patches), automação externa à instância do EC2 (como um aplicativo de gerenciamento de rede), outro AWS serviço (como AWS Systems Manager) ou uma reinicialização manual por uma pessoa.

O que fazer

Você pode verificar seus `slurmd` registros `slurmctld` ou para ver se sua instância foi reinicializada. Para obter mais informações, consulte Logs do agendador no AWS PCS e

Monitoramento de instâncias de AWS PCS usando a Amazon CloudWatch. O exemplo de entrada de `slurmctld` registro a seguir indica que a instância foi reinicializada:

Example

```
[2024-09-12T06:42:50.393+00:00] validate_node_specs: Node Login-1 unexpectedly rebooted
 boot_time=1726123354 last response=1726123285
```

Reinicializando devido à aplicação de patches

Geralmente, é necessária uma reinicialização após a aplicação dos patches. Não aplique patches diretamente a uma instância do EC2 que faz parte de um grupo de nós de computação do AWS PCS. Se você precisar corrigir suas instâncias do EC2, deverá aplicar seus patches a uma Amazon Machine Image (AMI) atualizada e atualizar seus grupos de nós de computação para usar a AMI atualizada. As novas instâncias do EC2 que o AWS PCS executa para esses grupos de nós de computação usarão a AMI atualizada (corrigida). Para obter mais informações, consulte Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

Solucionar problemas de inicialização e registro do nó de computação no PCS AWS

Quando os nós de computação não conseguem inicializar ou se registrar adequadamente em seu cluster AWS PCS, você pode enfrentar os seguintes sintomas:

- Os trabalhos não começam
- Você não pode se conectar às instâncias no AWS Systems Manager
- As instâncias foram encerradas inesperadamente
- As instâncias são substituídas continuamente

Essas falhas podem ser causadas por problemas durante a inicialização da instância EC2 ou durante o processo de inicialização do nó de computação do AWS PCS. Este tópico descreve procedimentos para ajudá-lo a solucionar problemas durante o processo de inicialização do nó AWS PCS. Para obter mais informações sobre como solucionar problemas de inicialização de instâncias do EC2, consulte Solucionar problemas de inicialização de instâncias do Amazon EC2 no Guia do usuário do Amazon Elastic Compute Cloud.

Falhas de bootstrap ocorrem quando uma instância do EC2 é iniciada com sucesso, mas falham durante o processo de ingresso no cluster AWS PCS. O processo de bootstrap inclui duas fases principais:

- Registro de nós — A instância do EC2 chama a ação da API RegisterComputeNodeGroupInstance AWS PCS para se registrar no serviço AWS PCS. Falhas podem ocorrer devido a problemas no seguinte:
  - Permissões
    - Perfil de instância errado
  - Redes
    - Não é possível conectar-se aos endpoints AWS PCS
    - Endpoint AWS PCS configurado incorretamente
    - Instância em uma sub-rede pública sem IP público
    - Instância multi-NIC em uma sub-rede pública
  - Segredo do cluster
    - O segredo do cluster foi excluído ou marcado para exclusão
- Integração do Slurm — A instância é executada `slurmd` e se junta ao cluster do Slurm. Falhas podem ocorrer devido a problemas no seguinte:
  - Permissões
    - Configuração do security group
    - Slurmctld não consegue executar ping no nó de computação
  - Configuração personalizada de AMI
    - Drivers NVIDIA ausentes
    - ResumeTimeout alcançado

Como o Slurm funciona no PCS AWS

Isso pode ajudá-lo a comparar a forma padrão de funcionamento do Slurm com a forma como o Slurm funciona no PCS. AWS

Processamento de trabalhos do Standard Slurm

As etapas a seguir ocorrem no processamento de tarefas padrão do Slurm:

1. Quando você envia um trabalho, `slurmctld` valida e coloca o trabalho em fila.

2. Quando os recursos se tornam disponíveis, `slurmctld` aloca os nós existentes.
3. `slurmd`daemons executam trabalhos em nós alocados.

Processamento de tarefas do Slurm no PCS AWS

As etapas a seguir ocorrem no processamento de tarefas do AWS PCS:

1. Quando você envia um trabalho, `slurmctld` valida e coloca o trabalho em fila.
2. Quando é necessária capacidade adicional, o AWS PCS usa o modelo de execução do grupo de nós de computação para iniciar novas instâncias do EC2.
3. Novas instâncias são inicializadas no cluster:
   a. As instâncias são registradas no AWS PCS.
   b. As instâncias se juntam ao cluster Slurm.
4. Quando os recursos estão prontos, `slurmctld` aloca os nós (incluindo os recém-inicializados).
5. `slurmd`daemons executam trabalhos em nós alocados.

Recuperar registros de instâncias

A primeira etapa para solucionar problemas de bootstrap do nó de computação é recuperar os registros da instância. É possível usar um dos seguintes métodos:

AWS CLI

Recupere a saída do console do nó de computação usando o seguinte comando:

```
aws ec2 get-console-output --region us-east-1 --instance-id i-1234567890abcdef0 --
output text
```

*us-east-1*Substitua pela sua AWS região e *i-1234567890abcdef0* pelo ID da sua instância.

AWS Systems Manager

Se você puder se conectar à instância usando o Systems Manager, poderá visualizar diretamente o arquivo de log do bootstrap:

1. Conecte-se à instância usando o Systems Manager. Para obter mais informações, consulte Iniciando uma sessão no Guia do Usuário do Systems Manager.
2. Veja o arquivo de log do bootstrap:

```
sudo cat /var/log/amazon/pcs/bootstrap.log
```

> **Note**
>
> Se houver um problema durante a fase de inicialização, talvez seja necessário esperar aproximadamente 20 minutos antes de se conectar à instância. Os serviços Systems Manager e SSH iniciam somente após a conclusão da inicialização ou quando a execução do bootstrap atinge um tempo limite em caso de falha.

Recuperar VPC/Subnet/Security grupos de um ID de instância

Para solucionar problemas com seus nós de computação, talvez seja necessário recuperar informações sobre a VPC, a sub-rede e os grupos de segurança associados às suas instâncias. Se você não conhece sua instância IDs, consulteEncontrando instâncias de grupos de nós de computação no AWS PCS.

Console de gerenciamento da AWS

Para obter VPC, sub-rede e grupos de segurança

1. Abra o console do Amazon EC2.
2. Selecione Instances (Instâncias).
3. Na tabela Instâncias, escolha o ID da instância.
4. Encontre o ID da VPC e o ID da sub-rede no resumo da instância exibido.
5. No resumo da instância, escolha a guia Segurança.
6. Encontre os grupos de segurança na guia Segurança.

AWS CLI

Use o comando a seguir para recuperar informações de VPC, sub-rede e grupo de segurança para sua instância:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
 'Reservations[*].Instances[*].
```

```
{InstanceId:InstanceId,VpcId:VpcId,SubnetId:SubnetId,SecurityGroups:SecurityGroups[*].GroupI
 --output table
```

Problemas de registro de nós

O registro do nó é a primeira ação executada por um nó de computação durante o bootstrap. O nó chama o endpoint da API AWS PCS para se registrar no AWS PCS. As falhas de registro geralmente mostram mensagens de erro semelhantes às seguintes:

```
<13>Nov  5 08:10:27 user-data: Recipe: aws-pcs-environment::node_registration
<13>Nov  5 08:10:27 user-data:   * ruby_block[Register NodeGroup Instance] action
 run[2024-11-05T08:10:27+00:00] INFO: Processing ruby_block[Register NodeGroup
 Instance] action run (aws-pcs-environment::node_registration line 19)
<13>Nov  5 08:15:46 user-data:
<13>Nov  5 08:15:46 user-data:
<13>Nov  5 08:15:46 user-data:
 ================================================================================
<13>Nov  5 08:15:46 user-data:     Error executing action `run` on resource
 'ruby_block[Register NodeGroup Instance]'
<13>Nov  5 08:15:46 user-data:
 ================================================================================
<13>Nov  5 08:15:46 user-data:
<13>Nov  5 08:15:46 user-data:     EOFError
```

Perfil de instância errado

Se a instância não conseguir se registrar, verifique se o perfil da instância associado ao nó de computação tem a `pcs:RegisterComputeNodeGroupInstance` permissão.

Para obter mais informações sobre como criar um perfil de instância válido, consulteCrie um perfil de instância para AWS PCS.

Não é possível conectar-se aos endpoints AWS PCS

Se seus nós de computação estiverem em uma sub-rede privada, verifique se você configurou endpoints VPC para AWS PCS ou se sua sub-rede tem uma rota para um gateway NAT para acesso à Internet. Para saber mais, consulte:

- Acesse um AWS serviço usando uma interface VPC endpoint no guia da Amazon Virtual Private Cloud. AWS PrivateLink

- Endpoints e cotas de serviço para PCS AWS.
- Conecte sua VPC a outras redes no Guia do usuário da Amazon Virtual Private Cloud
- AWS Rede PCS

Endpoint AWS PCS configurado incorretamente

Se você ver uma mensagem de erro semelhante à seguinte, verifique a política associada ao seu endpoint AWS PCS VPC:

```
com.amazon.coral.security.AccessDeniedException: User: arn:aws:sts::xxx:assumed-
role/rolename/i-instanceid is not authorized to perform:
 pcs:RegisterComputeNodeGroupInstance on resource: arn:aws:pcs:us-west-2:xxx:cluster/
cluster-id as either the resource does not exist, some policy explicitly denies access,
 or no policy grants access
```

Para obter mais informações sobre como configurar endpoints de interface VPC para AWS PCS, consulte. Acesso Serviço de Computação Paralela da AWS usando um endpoint de interface ()AWS PrivateLink

Instância em uma sub-rede pública sem IP público

Se sua sub-rede não tiver a atribuição automática de IP público habilitada e sua configuração de rota usar um gateway de internet, as instâncias não poderão se comunicar com a API AWS PCS.

As instâncias em uma sub-rede com um gateway de internet devem ter um endereço IP público. Para resolver esse problema, escolha uma das seguintes opções:

- Adicione um VPC endpoint para AWS PCS ao seu cluster VPC. Isso permite que as instâncias se comuniquem com o AWS PCS sem a necessidade de um endereço IP público passar pelo gateway da Internet.
- Use uma sub-rede privada com um gateway NAT, para que não seja necessário um endereço IP público.
- Ative a atribuição automática de endereços IP públicos por meio de sua sub-rede ou modelo de execução para que as instâncias possam entrar em contato com a API por meio do gateway da Internet. Observe que essa opção não é válida para instâncias de interface de várias redes.

Instância multi-NIC em uma sub-rede pública

Você deve usar uma sub-rede privada se usar um tipo de instância que tenha várias interfaces de rede (NICs).

AWS endereços IP públicos só podem ser atribuídos a instâncias iniciadas com uma única interface de rede. Para obter mais informações sobre endereços IP, consulte Atribuir um IPv4 endereço público durante a execução da instância no Guia do usuário do Amazon EC2 para instâncias Linux.

Os tipos de instância de várias NIC exigem um gateway NAT ou um proxy interno na sub-rede para acessar o AWS endpoint PCS. Como alternativa, você pode adicionar um VPC endpoint para AWS PCS ao seu cluster VPC.

O segredo do cluster foi excluído ou marcado para exclusão

Se o segredo compartilhado do Slurm no AWS Secrets Manager tiver sido excluído ou marcado para exclusão, os nós de computação não conseguirão se registrar e seu cluster ficará comprometido.

AWS O PCS cria automaticamente um segredo compartilhado do Slurm no AWS Secrets Manager (com formato de nome:`pcs!slurm-secret-<cluster-id>`) quando você cria um cluster. Esse segredo é necessário para comunicações seguras no cluster. Para obter mais informações, consulte Trabalhando com segredos de cluster no AWS PCS.

Se esse segredo for excluído ou marcado para exclusão, novos nós não poderão ingressar no cluster e o controlador ou outros daemons do cluster (como `slurmd` `eslurmdbd`) talvez não consigam se juntar novamente ao cluster se forem reiniciados.

Para resolver esse problema, você pode restaurar o segredo excluído se ele ainda estiver dentro da janela de recuperação. Para obter instruções detalhadas, consulte Restaurar um segredo do AWS Secrets Manager.

Se a janela de recuperação expirar, o segredo não poderá ser restaurado e o cluster AWS PCS afetado não poderá ser restaurado. Você precisa criar um novo cluster com a mesma configuração. AWS O PCS cria automaticamente um novo segredo do agendador.

Problemas de junção do cluster Slurm

Após o registro bem-sucedido do nó, o nó de computação tenta se juntar ao cluster Slurm. O `slurmd` daemon no nó entra em contato com o controlador Slurm para se registrar no cluster. As falhas de junção do Slurm geralmente mostram mensagens de erro semelhantes às seguintes:

```
<13>Nov  5 17:20:29 user-data: [2024-11-05T17:20:28+00:00] FATAL:
 Mixlib::ShellOut::ShellCommandFailed: service[slurmd] (aws-pcs-slurm::finalize_slurm
 line 18) had an error: Mixlib::ShellOut::ShellCommandFailed: Expected process to exit
 with [0], but received '1'
<13>Nov  5 17:20:29 user-data: ---- Begin output of ["/usr/bin/systemctl", "--system",
 "start", "slurmd"] ----
<13>Nov  5 17:20:29 user-data: STDOUT:
<13>Nov  5 17:20:29 user-data: STDERR: Job for slurmd.service failed because the
 control process exited with error code. See "systemctl status slurmd.service" and
 "journalctl -xe" for details.
<13>Nov  5 17:20:29 user-data: ---- End output of ["/usr/bin/systemctl", "--system",
 "start", "slurmd"] ----
```

Configuração do security group

Verifique se seus grupos de segurança estão configurados corretamente para permitir a comunicação entre os nós de computação e o controlador Slurm. Os grupos de segurança devem permitir o seguinte tráfego:

- Porta 6817 `slurmd` para comunicação com `slurmctld`
- Porta 6818 para `slurmctld` fazer ping `slurmd`

Para obter mais informações sobre os requisitos do grupo de segurança, consulte os tópicos a seguir:

- Crie grupos de segurança para AWS PCS
- Crie modelos de lançamento para AWS PCS
- Requisitos e considerações do grupo de segurança

> **Important**
>
> O grupo de segurança do cluster que você associou ao seu cluster durante a criação do cluster também deve ser configurado nos grupos de segurança do grupo de nós de computação para permitir que os nós de computação se comuniquem com o controlador.

Drivers NVIDIA ausentes

Se a instância for inicializada corretamente, mas os trabalhos não iniciarem e você ver mensagens de erro semelhantes às seguintes nos registros da instância, talvez você não tenha drivers da NVIDIA:

```
<13>Dec  2 13:52:00 user-data: [2024-12-02T13:52:00.094+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_config_always.sh: INFO: nvidia-smi not found!
...
<13>Dec  2 13:54:10 user-data: Job for slurmd.service failed because the control
 process exited with error code. See "systemctl status slurmd.service" and "journalctl
 -xe" for details.
<13>Dec  2 13:54:12 user-data: [2024-12-02T13:54:12.718+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_finalize.sh: INFO: systemctl could not start slurmd!
```

Se você se conectar à instância e verificar o status do `slurmd` daemon, poderá ver um erro semelhante ao seguinte:

```
$ systemctl status slurmd
...
fatal: can't stat gres.conf file /dev/nvidia0: No such file or directory
```

Para resolver esse problema, instale os drivers NVIDIA em sua AMI personalizada. Para obter mais informações, consulte Etapa 4 — (Opcional) Instale drivers, bibliotecas e software aplicativo adicionais.

ResumeTimeout alcançado

Se um nó de computação e sua instância do EC2 forem encerrados porque o nó não está íntegro, o AWS PCS pode não oferecer suporte à AMI ou pode haver problemas de rede. A instância do EC2 é executada por aproximadamente 30 minutos até que a do Slurm ResumeTimeout seja alcançada e marque o nó como. `DOWN`

Se a instância não inicializar corretamente e não estiver registrada no AWS PCS (nenhuma `RegisterComputeNodeGroupInstance` chamada para a instância EC2), verifique se há mensagens de erro semelhantes às seguintes nos registros da instância:

```
/opt/aws/pcs/bin/pcs_bootstrap_init.sh: No such file or directory
```

Esse erro indica que o software AWS PCS bootstrap não faz parte da AMI. Para resolver esse problema, certifique-se de que sua AMI personalizada inclua o software AWS PCS bootstrap. Para obter mais informações, consulte Imagens personalizadas da Amazon Machine (AMIs) para AWS PCS.

Slurmctld não consegue executar ping no nó de computação

Se a instância executar corretamente o procedimento de bootstrap e estiver registrada no AWS PCS, mas `slurmctld` não conseguir vê-la e enviar trabalhos para ela, a instância será configurada para `DOWN` depois de algum tempo e, em seguida, encerrada.

Isso pode ser causado por grupos de segurança configurados incorretamente. Por exemplo, se a porta 6817 estiver habilitada para permitir `slurmd` a comunicação`slurmctld`, mas a porta 6818 estiver ausente `slurmctld` para permitir o ping. `slurmd`

Verifique se seus grupos de segurança incluem todas as regras necessárias, conforme documentado emRequisitos e considerações do grupo de segurança.

Histórico de documentos do Guia do usuário do AWS PCS

A tabela a seguir descreve as mudanças importantes na documentação do AWS PCS.

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
10 de março de 2026	Agente PCS atualizado	Atualizado o tópico da AMI para o agente AWS PCS 1.3.2-1. Corrigido um problema que afetava o bootstrap do nó de computação do RHEL 8.10 e do Rocky Linux 8.10. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e AWS Versões do agente PCS.	N/D
11 de fevereiro de 2026	AWS PCS lançado na Ásia-Pacífico (Mumbai) e na Europa (Paris)	AWS O PCS agora está disponível na Ásia-Pacífico (Mumbai) (ap-south-1) e na Europa (Paris) (eu-west-3). CloudFormation modelos estão disponíveis para começar na Ásia-Pacífico (Mumbai) Região da AWS e na Europa (Paris) Região da AWS. Para obter mais informações, consulte Use CloudFormation para criar um cluster	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
		AWS PCS de amostra e CloudFormation modelos para criar um cluster AWS PCS de amostra.	
18 de novembro de 2025	Novo recurso: API REST do Slurm	A API REST do Slurm agora é compatível com o Slurm 25.05 ou posterior. Para obter mais informações, consulte API REST do Slurm em PCS AWS.	SDK DA AWS: 18/11/2025

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
10 de novembro de 2025	Novo recurso: suporte ao plug-in de filtro CLI do Slurm	AWS O PCS agora suporta plug-ins de filtro CLI do Slurm para executar scripts Lua personalizados que validam e modificam os parâmetros de envio de trabalhos antes que eles cheguem ao controlador Slurm. Use filtros CLI para aplicar políticas personalizadas, definir parâmetros padrão e fornecer orientação ao usuário durante o envio do trabalho. Esse recurso requer a versão 25.05 ou posterior do Slurm. Para obter mais informações, consulte Use os plug-ins de filtro CLI do Slurm para personalizar o envio de trabalhos no PCS AWS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
7 de novembro de 2025	Agente PCS atualizado	Atualizado o tópico da AMI para o agente AWS PCS 1.3.1-1. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e AWS Versões do agente PCS.	N/D
3 de novembro de 2025	Agente PCS atualizado e instaladores do Slurm	O tópico da AMI foi atualizado para o agente AWS PCS 1.3.0-1 e os instaladores do Slurm 24.11.6-2, 24.05.8-2 e 23.11.10-4. Lista atualizada de sistemas operacionais compatíveis. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e AWS Versões do agente PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
23 de outubro de 2025	Conteúdo atualizado: pcs-multi-cluster-login - configure.sh	Alguns erros foram corrigidos no script de configuração do nó de login de vários clusters. Para obter mais informações, consulte AWS Código de script de configuração do nó de login de vários clusters PCS.	N/D
21 de outubro de 2025	Novo recurso: rotação secreta do cluster	AWS O PCS agora oferece suporte à rotação secreta do cluster para aumentar a segurança . Para obter mais informações, consulte Segredos do cluster rotativo no AWS PCS. Permissões mínimas de administrador atualizadas para suportar a rotação secreta do cluster. Para obter mais informaçõ es, consulte Permissões mínimas para AWS PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
17 de outubro de 2025	Novo tópico: script de configuração de nós de login em vários clusters	Foi adicionado um novo tópico que fornece um script para configurar um nó de login independente para se conectar a vários clusters AWS PCS. O script automatiza a configuração de vários `sackd` daemons do Slurm e cria scripts de ativação para interação com o cluster. Para obter mais informações, consulte Conectando um nó de login independente a vários clusters no AWS PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
16 de outubro de 2025	Atualizado para o Slurm 25.05	Atualizou o guia do usuário para suporte ao Slurm 25.05. O Slurm 25.05 agora é a versão padrão. Para saber mais, consulte: • Versões Slurm no PCS AWS • Instaladores de software para criar de forma personalizada AMIs para AWS PCS • Notas de lançamento da amostra AWS PCS AMIs	N/D
16 de outubro de 2025	Agente PCS atualizado	Atualizado o tópico da AMI para o agente AWS PCS 1.2.2-1. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e AWS Versões do agente PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
2 de outubro de 2025	Novos recursos: reinicialização do Slurm node, atualizações de cluster e configurações personalizadas do Slurm	AWS O PCS adiciona suporte para vários novos recursos: • Reinicialização do Slurm node — Use o `scontrol reboot` comando nativo do Slurm para reinicializar os nós de computação sem a substituição da instância . Para obter mais informações, consulte Reinicializando nós de computação com o Slurm no PCS AWS. • Atualizações do cluster — modifique as configurações do cluster após a criação sem reconstruções. Para obter mais informações, consulte Atualizando um cluster no AWS PCS. • Configurações personalizadas do Slurm — defina parâmetros avançados do Slurm em recursos de cluster, fila e grupo de nós de computaçã	2025-10-01

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
		o. Para obter mais informações, consulte Definindo configurações personalizadas do Slurm no PCS AWS.	
23 de setembro de 2025	Novo tópico de solução de problemas: problemas de bootstrap do nó de computação	Foi adicionada orientação de solução de problemas para diagnosticar e resolver problemas de bootstrap do nó de computação. Para obter mais informações, consulte Solucionar problemas de inicialização e registro do nó de computação no PCS AWS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
17 de setembro de 2025	Novo recurso: blocos de capacidade para ML	AWS O PCS agora oferece suporte aos blocos de capacidade do Amazon EC2 para ML, que permitem que você reserve instâncias de computação acelerada baseadas em GPU para seus clusters. Para obter mais informações, consulte Usando blocos de capacidade do Amazon EC2 para ML com PCS AWS. As permissões mínimas para suportar blocos de capacidade agora fazem parte das permissões mínimas para um administrador de serviços. Para obter mais informações, consulte Permissões mínimas para AWS PCS.	2025-09-17

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
11 de setembro de 2025	Atualização da política gerenciada pela AWS	AWS O PCS atualizou o AWSPCSServiceRolePolicy para suportar blocos de capacidade. Para obter mais informações, consulte AWS políticas gerenciadas para o Serviço de Computação AWS Paralela.	N/D
14 de agosto de 2025	Documentação atualizada do perfil da instância	Aprimorou a documentação do perfil da instância com instruções abrangentes da CLI para criar funções e perfis de instância do IAM. Foram adicionados step-by-step procedimentos para configurar perfis de instância usando o AWS CLI e diretrizes aprimoradas para encontrar perfis de instância usados com o AWS PCS. Para obter mais informações, consulte Perfis de instância do IAM para o AWS Parallel Computing Service.	2025-08-14

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
1.º de agosto de 2025	Novo tópico: plugins SPANK	Foi adicionada documentação para plug-ins SPANK (Slurm Plug-in Architecture for Node and job Kontrol) que você pode usar para estender e modificar o comportamento do Slurm durante o lançamento e a execução do trabalho em clusters PCS. AWS Para obter mais informações, consulte Estenda a funcionalidade do Slurm no AWS PCS com plug-ins SPANK.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
1.º de agosto de 2025	IPv6 suporte de rede	Foi adicionado suporte para IPv6 rede ao criar clusters AWS PCS. Agora você pode escolher IPv6 o tipo de rede para seu cluster, com as atualizações correspondentes dos requisitos de VPC, configuração de sub-rede, configurações de grupos de segurança e procedimentos de criação de cluster. Para obter mais informações, consulte AWS Requisitos e considerações sobre PCS, VPC e sub-rede e Criando um cluster no AWS PCS.	2025-08-01

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
3 de julho de 2025	AWS PCS lançado na Europa (Londres)	AWS O PCS agora está disponível na Europa (Londres) (eu-west-2). CloudFormation modelos estão disponíveis para começar na Europa (Londres) Região da AWS. Para obter mais informações, consulte Use CloudFormation para criar um cluster AWS PCS de amostra e CloudFormation modelos para criar um cluster AWS PCS de amostra.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
1.º de julho de 2025	Instruções atualizadas do console	Agora você pode fazer com que o AWS PCS crie um perfil de instância básico e um grupo de segurança para você ao criar um cluster e um grupo de nós de computação no console. Para obter mais informações, consulte: • Criando um cluster no AWS PCS • Criação de um grupo de nós de computação no AWS PCS • Perfis de instância do IAM para o AWS Parallel Computing Service	N/D
23 de junho de 2025	Nova política gerenciada: AWSPCSComputeNodePolicy	Foi adicionada uma nova política gerenciada que concede permissão aos nós de computação do AWS PCS para se conectarem aos clusters do AWS PCS. Para obter mais informações, consulte AWS política gerenciada: AWSPCSComputeNodePolicy.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
19 de junho de 2025	Novo tópico: registros de conclusão do trabalho	Use registros de conclusão do trabalho para registrar detalhes sobre os trabalhos quando eles forem concluídos, sem custo adicional. Para obter mais informações, consulte Registros de conclusão de trabalhos no AWS PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
18 de junho de 2025	AWS Lançamento do PCS em AWS GovCloud (US)	AWS O PCS agora está disponível em AWS GovCloud (Leste dos EUA) (us-gov-east-1) e AWS GovCloud (Oeste dos EUA) (us-gov-west-1). CloudFormation modelos estão disponíveis para começar no AWS GovCloud (US) Regions. Para obter mais informações, consulte Use CloudFormation para criar um cluster AWS PCS de amostra e CloudFormation modelos para criar um cluster AWS PCS de amostra. Para obter mais informações sobre os endpoints do serviço AWS PCS em AWS GovCloud (US) Regions, consulteEndpoints e cotas de serviço para PCS AWS. Para obter mais informações sobre as diferenças em AWS GovCloud (US) Regions,	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
		consulte AWS PCS AWS GovCloud (US) no Guia AWS GovCloud (US) do Usuário.	
18 de junho de 2025	Agente PCS atualizado	Atualizado o tópico da AMI para o agente AWS PCS 1.2.1-1. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS.	N/D
15 de maio de 2025	Novo recurso: contabilidade	A contabilidade do Slurm agora é compatível com o Slurm 24.11 ou posterior. Para obter mais informações, consulte Contabilidade de slurm no PCS AWS.	SDK DA AWS: 15/05/2015

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
15 de maio de 2025	Atualizado para o Slurm 24.11	Atualizou o guia do usuário para suporte ao Slurm 24.11.5. Para saber mais, consulte: • Versões Slurm no PCS AWS • Instaladores de software para criar de forma personalizada AMIs para AWS PCS • Notas de lançamento da amostra AWS PCS AMIs	N/D
5 de maio de 2025	Perguntas frequentes sobre as versões atualizadas do Slurm	Perguntas frequentes (FAQ) das versões do Slurm atualizadas sobre versões do Slurm próximas ou além do fim da vida útil (EOL). Para obter mais informações, consulte Perguntas frequentes sobre as versões do Slurm no PCS AWS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
17 de abril de 2025	Novo tópico: como obter detalhes do grupo de nós de computação	Saiba como obter detalhes de um grupo de nós de computação do AWS PCS, como ID, ARN e ID de AMI. Para obter mais informações, consulte Obtenha detalhes do grupo de nós de computação no AWS PCS.	N/D
2 de abril de 2025	Instalador Slurm atualizado	Atualizado o tópico da AMI para o instalador do Slurm 24.05.7-1. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS.	N/D
28 de março de 2025	Foram adicionadas cotas para o número máximo de grupos e filas de nós de computação	Foram adicionadas cotas internas não ajustáveis para o número máximo de grupos de nós de computação por cluster e o número máximo de filas por cluster. Para obter mais informações, consulte Cotas internas.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
14 de março de 2025	Alterou uma chave de propriedade no CloudFormation modelo	`Id` agora é `TemplateId` para a `CustomLaunchTemplate` propriedade no CloudFormation modelo. Para obter mais informações, consulte Recursos em Partes de um CloudFormation modelo para AWS PCS.	N/D
13 de março de 2025	Informações de versão adicionadas para o agente AWS PCS e o Slurm	Foi adicionado um novo tópico que descreve as alterações em cada versão do agente AWS PCS. Para obter mais informações, consulte AWS Versões do agente PCS. Foram adicionadas mais informações ao tópico de versões do Slurm que descreve datas de suporte importantes e notas de lançamento detalhadas para o suporte do AWS PCS para o Slurm. Para obter mais informações, consulte Versões Slurm no PCS AWS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
07 de março de 2025	Agente PCS atualizado	Atualizado o tópico da AMI para o agente AWS PCS 1.2.0-1. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS.	N/D
3 de fevereiro de 2025	Foi adicionado um tópico sobre o uso AWS CloudFormation com o AWS PCS	Foi adicionado um tópico ao guia do usuário que fornece um exemplo de como usar CloudFormation com o AWS PCS. O tópico fornece um procedimento para usar um CloudFormation modelo de amostra para criar o cluster AWS PCS de amostra e descreve resumidamente as seções desse modelo. Para obter mais informações, consulte Comece a usar um CloudFormation AWS PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
18 de dezembro de 2024	Atualizado para o Slurm 24.05	Atualizou o guia do usuário para suporte ao Slurm 24.05. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e Notas de lançamento da amostra AWS PCS AMIs.	N/D
18 de dezembro de 2024	Versões atualizadas da NVIDIA para a amostra Slurm 23.11 AMIs	O driver NVIDIA e as versões CUDA foram atualizados na amostra do Slurm 23.11. AMIs Para obter mais informações, consulte Notas de lançamento da amostra AWS PCS AMIs.	N/D
17 de dezembro de 2024	Instalador Slurm atualizado	Atualizado o tópico da AMI para o instalador do Slurm 23.11.10-3. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
13 de dezembro de 2024	Agente PCS atualizado	Atualizado o tópico da AMI para o agente AWS PCS 1.1.1-1. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS.	N/D
06 de dezembro de 2024	Agente PCS atualizado e instalador do Slurm	O tópico da AMI foi atualizado para o agente AWS PCS 1.1.0-1 e o instalador do Slurm 23.11.10-2. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS.	N/D
06 de dezembro de 2024	Foi adicionado um tópico sobre suporte ao sistema operacional	Para obter mais informações, consulte Sistemas operacionais compatíveis no AWS PCS.	N/D
8 de novembro de 2024	Guia do usuário reorganizado	Reorganizamos o guia do usuário para colocar os tópicos no nível superior, movemos alguns tópicos para suas próprias páginas e agrupamos tópicos semelhantes.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
7 de novembro de 2024	Tópicos atualizados da AMI	Atualizado o tópico da AMI para o Slurm 23.11.10 e libjwt 17.0. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e Etapa 3 — Instalar o Slurm. Simplificou e corrigiu as notas de lançamento do. AMIs Para obter mais informações, consulte Notas de lançamento da amostra AWS PCS AMIs.	N/D
7 de novembro de 2024	Foi adicionado um novo tópico sobre o uso de volumes criptografados do EBS com AWS o PCS	Foi adicionado um tópico que descreve a política de chaves do KMS necessária para volumes criptografados do EBS no AWS PCS. Para obter mais informações, consulte Política de chave KMS necessária para uso com volumes criptografados do EBS no PCS AWS.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
18 de outubro de 2024	AWS Lançado o agente PCS 1.0.1-1	Documentação relacionada à AMI atualizada para se referir à versão 1.0.1-1 do agente AWS PCS. Para obter mais informações, consulte Instaladores de software para criar de forma personalizada AMIs para AWS PCS e Etapa 2 — Instalar o agente AWS PCS.	N/D
10 de outubro de 2024	Foi adicionado um capítulo de solução de problemas	Foi adicionado um capítulo de solução de problemas com um tópico sobre a substituição automática de instâncias do EC2 após uma reinicialização. Para obter mais informações, consulte Solução de problemas no serviço de computação AWS paralela.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
23 de setembro de 2024	Atualizou as permissões mínimas para usar ações de API e para um administrador de serviços	Agora, a `ec2:DescribeInstanceTypeOfferings` permissão é necessária para `CreateComputeNodeGroup` as ações `UpdateComputeNodeGroup` da API. Para obter mais informações, consulte Permissões mínimas para AWS PCS.	N/D
5 de setembro de 2024	Atualizou o exemplo de política do IAM para as permissões mínimas para um administrador de serviços	Para obter mais informações, consulte Permissões mínimas para um administrador de serviços.	N/D
5 de setembro de 2024	Foi adicionada uma permissão ausente ao JSON na página de políticas gerenciadas	Essa foi apenas uma correção na documentação. A política gerenciada real não foi alterada. Para obter mais informações, consulte AWS políticas gerenciadas para o Serviço de Computação AWS Paralela.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
28 de agosto de 2024	Página de políticas gerenciadas adicionada	Para obter mais informações, consulte AWS políticas gerenciadas para o Serviço de Computação AWS Paralela.	N/D
28 de agosto de 2024	AWS Lançamento do PCS	Versão inicial do guia do usuário do AWS PCS.	AWS SDK: 2024-08-28

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário na Glossário da AWS Referência.

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.