aws

用户指南

AWS PC

AWS PC: 用户指南

Table of Contents

什么是 AWS 并行计算服务？

AWS Parallel Computing Service (AWS PCS) 是一项托管服务，它可以更轻松地运行和扩展高性能计算 (HPC) 工作负载，并在 AWS 使用 Slurm 的基础上构建科学和工程模型。 AWS PCS用于构建集同类最佳计算、存储、网络和可视化于一体的 AWS 计算集群。运行仿真或构建科学和工程模型。使用内置的管理和可观察性功能简化和简化集群操作。让您的用户能够在熟悉的环境中运行应用程序和作业，从而使他们能够专注于研究和创新。

主题

中的概念 AWS PCS

中的一个集群 AWS PCS有 1 个或多个队列，这些队列与至少 1 个计算节点组相关联。作业提交到队列并在计算节点组定义的EC2实例上运行。您可以使用这些基础来实现复杂的HPC架构。

集群

集群是一种用于管理资源和运行工作负载的资源。集群是一种 AWS PCS资源，用于定义计算、网络、存储、身份和作业调度器配置的组合。您可以通过指定要使用哪个作业调度程序（当前为 Slurm）、想要的调度器配置、要管理集群的服务控制器以及要在其中VPC启动集群资源来创建集群。调度器接受和调度作业，还启动处理这些任务的计算节点（EC2实例）。

计算节点组

计算节点组是计算节点的集合， AWS PCS用于运行作业或提供对集群的交互式访问。在定义计算节点组时，您需要指定常见特征，EC2例如 Amazon 实例类型、最小和最大实例数、目标VPC子网、Amazon 系统映像 (AMI)、购买选项和自定义启动配置。 AWS PCS使用这些设置来高效启动、管理和终止计算节点组中的计算节点。

队列

当你想在特定集群上运行作业时，可以将其提交到特定的队列（有时也称为分区）。该作业将一直保留在队列中，直到 AWS PCS安排它在计算节点组上运行。您可以将一个或多个计算节点组与每个队列相关联。需要一个队列才能使用作业调度器提供的各种调度策略在底层计算节点组资源上调度和执行作业。用户不会直接向计算节点或计算节点组提交作业。

系统管理员

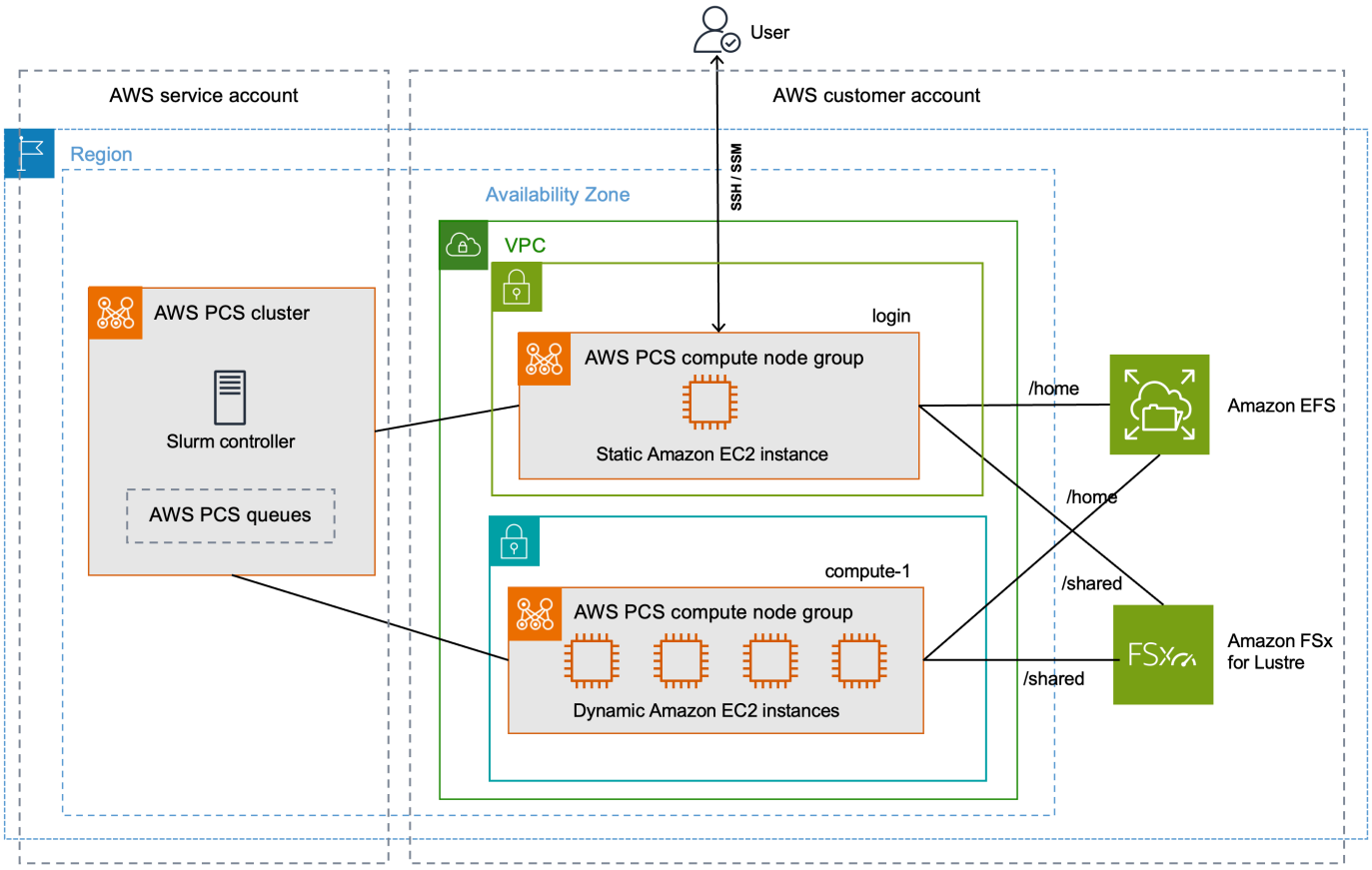
系统管理员部署、维护和操作集群。他们可以 AWS PCS通过 AWS Management Console AWS PCSAPI、和进行访问 AWS SDK。他们可以通过SSH或访问特定的集群 AWS Systems Manager，在那里他们可以运行管理任务、运行作业、管理数据以及执行其他基于 shell 的活动。有关更多信息，请参阅 AWS Systems Manager 文档。

最终用户

最终用户没有 day-to-day责任部署或操作集群。他们使用终端接口（例如SSH）来访问群集资源、运行作业、管理数据和执行其他基于 shell 的活动。

AWS 并行计算服务入门

这是一个创建简单集群的教程，你可以用它来试用 AWS PCS。下图显示了集群的设计。



集群设计教程包含以下关键组件：

- A VPC 和符合AWS PCS网络要求的子网。
- Amazon EFS 文件系统，将用作共享的主目录。
- Amazon FSx for Lustre 文件系统，它提供共享的高性能目录。
- 一个 AWS PCS集群，它提供 Slurm 控制器。
- 2 个 AWS PCS计算节点组。
  - `login`节点组，它提供对系统的基于 shell 的交互式访问。
  - `compute-1`节点组提供弹性伸缩实例来运行作业。
- 1 个向`compute-1`节点组中的EC2实例发送任务的队列。

集群需要其他 AWS 资源，例如安全组、IAM角色和EC2启动模板，这些资源未显示在图表中。

> **Note**
>
> 我们建议您在 Bash shell 中完成本主题中的命令行步骤。如果您没有使用 Bash shell，则某些脚本命令（例如行延续字符以及变量的设置和使用方式）需要调整 shell。此外，您的 Shell 的引用和转义规则可能有所不同。有关更多信息，请参阅《版本 2 AWS Command Line Interface 用户指南》 AWS CLI中的 “引号和带字符串的文字”。

主题

入门的先决条件 AWS PCS

请参阅以下主题，为你 AWS 账户 和本地的开发环境做好准备 AWS PCS。

主题

注册 AWS 并创建管理员用户

完成以下任务以设置 AWS 并行计算服务 (AWS PCS)。

主题

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

要注册 AWS 账户

1. 打开https://portal.aws.amazon.com/billing/注册。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。您可以随时前往 https://aws.amazon.com/并选择“我的账户”，查看您当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的以根用户身份登录。
2. 为您的 root 用户开启多重身份验证 (MFA)。

有关说明，请参阅《用户指南》中的 “为 AWS 账户 root 用户（控制台）启用虚拟MFA设备” IAM。

创建具有管理访问权限的用户

1. 启用IAM身份中心。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的启用 AWS IAM Identity Center。

2. 在 IAM Identity Center 中，向用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 Ident IAM ity Center 用户登录URL，请使用您在创建 Ident IAM ity Center 用户时发送到您的电子邮件地址的登录信息。

  有关使用 Ident IAM ity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个遵循应用最低权限权限的最佳实践的权限集。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的创建权限集。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的添加组。

安装 AWS CLI

您必须使用最新版本的 AWS CLI。有关信息，请参阅版本 2 AWS Command Line Interface 用户指南 AWS CLI中的安装或更新到最新版本的。

您必须配置 AWS CLI。有关更多信息，请参阅版本 2 AWS Command Line Interface 用户指南 AWS CLI中的配置。

在命令提示符下输入以下命令进行检查 AWS CLI；它应该会显示帮助信息。

```
aws pcs help
```

所需的IAM权限 AWS PCS

您使用的IAM安全主体必须具有使用 AWS PCSIAM角色、服务关联角色、 AWS CloudFormation VPC、和相关资源的权限。有关更多信息并 AWS 行计算服务的 Identity and Access 管理，请参阅AWS Identity and Access Management 用户指南中的和创建服务相关角色。您必须以同一用户身份完成本指南中的所有步骤。要查看当前用户，请运行以下命令：

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

为创建VPC和子网 AWS PCS

您可以使用 CloudFormation 模板创建VPC和子网。使用以下命令URL下载 CloudFormation 模板，然后在AWS CloudFormation 控制台中上传模板以创建新 CloudFormation堆栈。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的使用 AWS CloudFormation 控制台。

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

在 AWS CloudFormation 控制台中打开模板后，输入以下选项。您可以使用模板中提供的默认值。

- 在 “提供堆栈名称” 下：
  - 在堆栈名称下，输入：

    ```
    hpc-networking
    ```
- 在 “参数” 下：
  - 下方 VPC：
    - 在下CidrBlock方输入：

      ```
      10.3.0.0/16
      ```
  - 在子网 A 下：
    - 在 CidrPublicSubnetA 下，输入：

```
10.3.0.0/20
```

- 在 CidrPrivateSubnetA 下，输入：

```
10.3.128.0/20
```

- 在子网 B 下：
  - 在 CidrPublicSubnetB 下，输入：

```
10.3.16.0/20
```

  - 在 CidrPrivateSubnetB 下，输入：

```
10.3.144.0/20
```

- 在子网 C 下：
  - 对于 ProvisionSubnetsC，选择 True
  - 在 CidrPublicSubnetC 下，输入：

```
10.3.32.0/20
```

  - 在 CidrPrivateSubnetC 下，输入：

```
10.3.160.0/20
```

- 在 “能力” 下：
  - 选中 “我确认这 AWS CloudFormation 可能会创建IAM资源” 复选框。

监控 CloudFormation 堆栈的状态。当它到达时`CREATE_COMPLETE`，在新安全组中找到默认安全组的 ID VPC。您将在本教程的后面部分使用该 ID。

查找集群的默认安全组 VPC

要在新版中查找默认安全组的 IDVPC，请按照以下步骤操作：

- 导航至 Amazon VPC 控制台。
- 在 “VPC控制面板” 下，选择 “筛选依据” VPC。
  - 选择名称VPC的开头`hpc-networking`。

- 在 “安全” 下，选择 “安全组”。
- 查找名为的组的安全组 ID `default`。它有描述`default VPC security group`。您可以稍后使用该 ID 来配置EC2启动模板。

为创建安全组 AWS PCS

AWS PCS依靠安全组来管理进出集群及其计算节点组的网络流量。有关此主题的详细信息，请参阅安全组要求和注意事项。

在此步骤中，您将使用 CloudFormation 模板创建两个安全组。

- 集群安全组，它支持 AWS PCS控制器、计算节点和登录节点之间的通信。
- 入站SSH安全组，您可以选择将其添加到登录节点以支持SSH访问

为创建安全组 AWS PCS

您可以使用 CloudFormation 模板来创建安全组。使用以下命令URL下载 CloudFormation 模板，然后在AWS CloudFormation 控制台中上传模板以创建新 CloudFormation 堆栈。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的使用 AWS CloudFormation 控制台。

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

在 AWS CloudFormation 控制台中打开模板后，输入以下选项。请注意，某些选项将在模板中预先填充，您只需将其保留为默认值即可。

- 在 “提供堆栈名称” 下
  - 在堆栈名称下，输入：

    ```
    getstarted-sg
    ```
- 在 “参数” 下
  - 在下方 VpcId，选择名称的开头VPC位置`hpc-networking`。
  - （可选）在下方 ClientIpCidr，为入站SSH安全组输入限制性更强的 IP 范围。我们建议您使用自己的 IP/子网对此进行限制（x.x.x.x/32 代表您自己的 IP，x.x.x.x/24 表示范围。 将 x.x.x.x 替换为你自己的 IP。PUBLIC 你可以使用诸如 https://ifconfig.co/ 之类的工具获取你的公有 IP)

监控 CloudFormation 堆栈的状态。当它到达安全CREATE_COMPLETE组时，资源就准备好了。

创建了两个安全组，名称为：

- cluster-getstarted-sg— 这是群集安全组
- inbound-ssh-getstarted-sg— 这是一个允许入站SSH访问的安全组

在中创建集群 AWS PCS

在中 AWS PCS，群集是一种永久性资源，用于管理资源和运行工作负载。您可以在新的或现有的子网中为特定的调度程序（AWS PCS当前支持 Slurm）创建集群。VPC集群接受和调度作业，还会启动处理这些任务的计算节点（EC2实例）。

创建集群的步骤

1. 打开AWS PCS控制台并选择创建集群。
2. 在集群详细信息部分，输入以下字段：
   - 集群名称-输入 get-started
   - 调度程序 — 选择 Slur m 版本 24.05
   - 控制器大小-选择小
3. 在“网络”部分中，为以下字段选择值：
   - VPC— 选择VPC名字的 hpc-networking:Large-Scale-HPC
   - 子网-选择名称开头的子网 hpc-networking:PrivateSubnetA
   - 安全组-选择名为的集群安全组 cluster-getstarted-sg
4. 选择创建集群。

> **Note**
>
> 置备集群时，状态字段显示正在创建。创建集群可能需要几分钟。

在 Amazon Elastic File System AWS PCS 中创建共享存储

Amazon Elastic File System (AmazonEFS) 是一项提供无服务器、完全弹性的文件存储的 AWS 服务，因此您无需预置或管理存储容量和性能即可共享文件数据。有关更多信息，请参阅 Amazon Elastic File System 用户指南中的什么是 Amazon Elastic File System？。

AWS PCS演示集群使用EFS文件系统在群集节点之间提供共享的主目录。在与您的集群VPC相同的EFS文件系统中创建文件系统。

创建您的 Amazon EFS 文件系统

1. 前往 Amazon EFS 控制台。
2. 确保将其设置为与您要尝试的 AWS 区域 位置相同 AWS PCS。
3. 选择创建文件系统。
4. 在创建文件系统页面上，设置以下参数：
   - 对于名称，输入 `getstarted-efs`。
   - 在虚拟私有云 (VPC) 下，选择VPC命名的 `hpc-networking:Large-Scale-HPC`
   - 选择创建。这会将您返回到 “文件系统” 页面。
5. 记下文件系统的**`getstarted-efs`**文件系统 ID。稍后您将使用此信息。

在 Amazon AWS PCS FSx 中为 Lustre 创建共享存储空间

Amazon f FSx or Lustre 可以轻松且经济高效地启动和运行流行的高性能 Lustre 文件系统。您可以将 Lustre 用于速度至关重要的工作负载，例如机器学习、高性能计算 (HPC)、视频处理和财务建模。有关更多信息，请参阅 Amazon FSx for Lustre 是什么？ 在《Amazon f FSx or Lustre 用户指南》中。

AWS PCS演示集群可以使用 f FSx or Lustre 文件系统在群集节点之间提供高性能共享目录。在与您的集群相同FSxVPC的 for Lustre 文件系统中创建。

创建 for L FSx ustre 文件系统

1. 前往 Amazon FSx 控制台。
2. 确保将控制台设置为使用与您的集群 AWS 区域 相同的控制台。
3. 选择创建文件系统。
   - 在 “选择文件系统类型” 中，选择 Amazon f FSx or Lustre，然后选择 “下一步”。

4. 在指定文件系统详细信息页面上，设置以下参数：
   - 在 “文件系统详细信息” 下
     - 对于名称，输入 `getstarted-fsx`。
     - 对于部署和存储类型，选择持久，SSD
     - 对于每单位存储的吞吐量，请选择 125 MB /s/TiB
     - 对于存储容量，请输入 1.2 TiB
     - 在 “元数据配置” 中，选择 “自动”
     - 对于数据压缩类型，选择 LZ4
   - 在 “网络与安全” 下
     - 对于虚拟私有云 (VPC)，请选择VPC命名的 `hpc-networking:Large-Scale-HPC`
     - 对于VPC安全组，请将安全组命名为 `default`
     - 对于子网，请选择名称开头的子网 `hpc-networking:PrivateSubnetA`
   - 将其他选项设置为其默认值。
   - 选择下一步。
5. 在 “查看并创建” 页面上，选择 “创建文件系统”。这将使您返回到文件系统页面。
6. 导航到您创建的 for Lustre 文件系统的详细信息页面。FSx
7. 记下文件系统 ID 和装载名称。稍后您将使用此信息。

> **Note**
>
> “状态” 字段显示在置备文件系统时正在创建。创建文件系统可能需要几分钟。等到它完成后再继续本教程的其余部分。

在中创建计算节点组 AWS PCS

计算节点组是 AWS PCS启动和管理的计算节点（EC2实例）的虚拟集合。在定义计算节点组时，需要指定常见特征，例如EC2实例类型、最小和最大实例数、目标VPC子网、首选购买选项和自定义启动配置。 AWS PCS根据这些设置，有效地启动、管理和终止计算节点组中的计算节点。演示集群使用计算节点组为用户访问提供登录节点，使用单独的计算节点组来处理作业。以下主题描述了在集群中设置这些计算节点组的过程。

主题

- 为创建实例配置文件 AWS PCS
- 为创建启动模板 AWS PCS
- 在中为登录节点创建计算节点组 AWS PCS
- 创建用于在中运行计算作业的计算节点组 AWS PCS

为创建实例配置文件 AWS PCS

计算节点组在创建时需要实例配置文件。如果您使用为 Amazon 创建角色EC2，则控制台会自动创建实例配置文件并将其命名为与角色相同的名称。 AWS Management Console 有关更多信息，请参阅AWS Identity and Access Management 用户指南中的使用实例配置文件。

在以下步骤中，您可以使用为 Amazon 创建角色EC2，该角色还会为您的计算节点组创建实例配置文件。 AWS Management Console

创建角色和实例配置文件

- 导航到 IAM 控制台。
- 在访问管理下，选择策略。
  - 选择 Create policy (创建策略)。
  - 在 “指定权限” 下的 “策略编辑器” 中，选择JSON。
  - 将文本编辑器的内容替换为以下内容：

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

  - 选择下一步。
  - 在 “查看并创建” 下，在 “策略名称” 中输入AWSPCS-getstarted-policy。
  - 选择创建策略。

- 在 Access management（访问管理）下，请选择 Roles（角色）。
- 选择 Create role（创建角色）。
- 在“选择可信实体”下：
  - 对于“可信实体类型”，选择“AWS 服务”
  - 在“用例”下，选择EC2。
    - 然后，在“为指定服务选择用例”下，选择EC2。
  - 选择下一步。
- 在“添加权限”下：
  - 在权限策略中，搜索 AWSPCS-getstarted-policy。
  - 选中 AWSPCS-getstarted-policy 旁边的复选框将其添加到角色中。
  - 在权限策略中，搜索 A mazonSSMManaged InstanceCore。
  - 选中 A 旁边的复选框将其mazonSSMManagedInstanceCore添加到角色中。
  - 选择下一步。
- 在“名称”下，查看并创建：
  - 在“角色详情”下：
    - 对于 Role name（角色名称），输入 `AWSPCS-getstarted-role`。
  - 请选择 Create role（创建角色）。

为创建启动模板 AWS PCS

创建计算节点组时，您需要提供一个 AWS PCS用于配置其EC2启动的EC2实例的启动模板。这包括实例启动时运行的安全组和脚本等设置。

在此步骤中，将使用一个 CloudFormation 模板来创建两个EC2启动模板。一个模板将用于创建登录节点，另一个模板将用于创建计算节点。它们之间的关键区别在于，可以将登录节点配置为允许入站SSH访问。

访问 CloudFormation 模板

使用以下命令URL下载 CloudFormation 模板，然后在AWS CloudFormation 控制台中上传模板以创建新 CloudFormation 堆栈。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的使用 AWS CloudFormation 控制台。

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

使用 CloudFormation 模板创建EC2启动模板

使用以下步骤在 AWS CloudFormation 控制台中完成 CloudFormation 模板

- 在 “提供堆栈名称” 下：
  - 在堆栈名称下，输入`getstarted-lt`。
- 在 “参数” 下：
  - 在安全之下
    - 对于 VpcSecurityGroupId，选择集群`default`中命名的安全组VPC。
    - 对于 ClusterSecurityGroupId，请选择名为的群组 `cluster-getstarted-sg`
    - 对于 SshSecurityGroupId，请选择名为的群组 `inbound-ssh-getstarted-sg`
    - 对于 SshKeyName，请选择您的首选SSH密钥对。
  - 在 “文件系统” 下
    - 对于 EfsFilesystemId，请输入您在本教程前面创建EFS的文件系统的文件系统 ID。
    - 对于 FSxLustreFilesystemId，请输入您在本教程前面创建FSx的 for Lustre 文件系统的文件系统 ID。
    - 对于 FSxLustreFilesystemMountName，输入 Lustre 文件FSx系统的相同装载名称。
- 选择 “下一步”，然后再次选择 “下一步”。
- 选择提交。

监控 CloudFormation 堆栈的状态。当它到达`CREATE_COMPLETE`时，启动模板就可以使用了。

> **Note**
>
> 要查看 CloudFormation 模板创建的所有资源，请打开AWS CloudFormation 控制台。选择 `getstarted-lt` 堆栈，然后选择 Resources（资源）选项卡。

在中为登录节点创建计算节点组 AWS PCS

计算节点组是 AWS PCS启动和管理的计算节点（EC2实例）的虚拟集合。在定义计算节点组时，需要指定常见特征，例如EC2实例类型、最小和最大实例数、目标VPC子网、首选购买选项和自定义启动配置。 AWS PCS根据这些设置，有效地启动、管理和终止计算节点组中的计算节点。

在此步骤中，您将启动一个提供集群交互式访问权限的静态计算节点组。你可以使用SSH或 Amazon S EC2 ystems Manager (SSM) 登录它，然后运行 shell 命令并管理 Slurm 作业。

创建计算节点组

- 打开AWS PCS控制台并导航到集群。
- 选择名为的集群 `get-started`
- 导航到 “计算节点组”，然后选择 “创建”。
- 在计算节点组设置部分，提供以下内容：
  - 计算节点组名称-输入`login`。
- 在 “计算配置” 下，输入或选择以下值：
  - EC2启动模板-选择名称所在的启动模板 `login-getstarted-lt`
  - IAM实例配置文件-选择名为的实例配置文件 `AWSPCS-getstarted-role`
  - 子网-选择名称开头的子网。`hpc-networking:PublicSubnetA`
  - 实例-选择`c6i.xlarge`。
  - 扩展配置-对于最小实例数，请输入1。在 “最大实例数” 中，输入**1**。
- 在 “其他设置” 下，指定以下内容：
  - AMIID — 选择AMI要使用的名称，其格式如下：

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

    有关该示例的更多信息AMIs，请参阅使用示例 Amazon 系统映像 (AMIs) AWS PCS。
- 选择创建计算节点组。

在配置计算节点组时，状态字段显示正在创建。在本教程的下一步中，您可以继续进行下一步。

创建用于在中运行计算作业的计算节点组 AWS PCS

在此步骤中，您将启动一个计算节点组，该节点组可以弹性扩展以运行提交到集群的作业。

创建计算节点组

- 打开AWS PCS控制台并导航到集群。
- 选择名为的集群 `get-started`
- 导航到 “计算节点组”，然后选择 “创建”。
- 在计算节点组设置部分，提供以下内容：
  - 计算节点组名称-输入`compute-1`。
- 在 “计算配置” 下，输入或选择以下值：
  - EC2启动模板-选择名称所在的启动模板 `compute-getstarted-lt`
  - IAM实例配置文件-选择名为的实例配置文件 `AWSPCS-getstarted-role`
  - 子网-选择名称开头的子网。`hpc-networking:PrivateSubnetA`
  - 实例-选择`c6i.xlarge`。
  - 扩展配置-对于最小实例数，请输入`0`。在最大实例数中，输入`4`。
- 在 “其他设置” 下，指定以下内容：
  - AMIID — 选择AMI要使用的名称，其格式如下：

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

    有关该示例的更多信息AMIs，请参阅使用示例 Amazon 系统映像 (AMIs) AWS PCS。
- 选择创建计算节点组。

在配置计算节点组时，状态字段显示正在创建。

> **Important**
>
> 等待 “状态” 字段显示为 “活动”，然后再继续本教程的下一步。

创建队列来管理作业 AWS PCS

您可以将作业提交到队列以运行该任务。该作业将一直保留在队列中，直到 AWS PCS安排它在计算节点组上运行。每个队列都与一个或多个计算节点组相关联，这些节点组提供了执行处理所需的EC2实例。

在此步骤中，您将创建一个使用计算节点组处理作业的队列。

创建队列

- 打开控制AWS PCS台。
- 选择名为 get-started 的集群。
- 导航到 “计算节点组”，并确保该compute-1组的状态为 “活动”。

> **Important**
>
> 在继续下一步之前，compute-1群组的状态必须为 “活动”。

- 导航到 “队列”，然后选择 “创建队列”。
  - 在队列配置部分中，提供以下值：
    - 队列名称-输入以下内容：demo
    - 计算节点组-选择名为的计算节点组compute-1。
- 选择创建队列。

创建队列时，状态字段显示正在创建。

> **Important**
>
> 等待 “状态” 字段显示为 “活动”，然后再继续本教程的下一步。

Connect 连接到您的 AWS PCS集群

login计算节点组的状态变为 “活动” 后，您可以连接到它创建的EC2实例。

连接到登录节点

- 打开AWS PCS控制台并导航到集群。
- 选择名为 get-started 的集群。
- 选择计算节点组。
- 导航到名为的计算节点组login。
- 找到计算节点组 ID。
- 在另一个浏览器窗口或选项卡中，打开 Amazon EC2 控制台。
  - 选择实例。

- 搜索带有以下标签的EC2实例。Replace（替换）*node-group-id* 使用上一步中的计算节点组 ID 的值。应该有 1 个实例。

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

- Connect 连接到EC2实例。您可以使用会话管理器或SSH。

  Session Manager

  - 选择实例。
  - 选择连接。
  - 在 Connect to 实例下，选择会话管理器。
  - 选择连接。
  - 选择连接。交互式终端将在您的浏览器中启动。

  SSH

  - 选择实例。
  - 选择连接。
  - 在 Connect to 实例下，选择SSH客户端。
  - 按照控制台提供的说明进行操作。

  > **Note**
  >
  > 该实例的用户名**ec2-user**不是root。

在中探索集群环境 AWS PCS

登录到集群后，您可以运行 shell 命令。例如，您可以更改用户、处理共享文件系统上的数据以及与 Slurm 交互。

更改用户

如果您使用会话管理器登录到集群，则可能以身份进行连接ssm-user。这是为会话管理器创建的特殊用户。使用以下命令在 Amazon Linux 2 上切换到默认用户。如果您使用连接，则无需执行此操作 SSH。

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

使用共享文件系统

您可以使用命令确认EFS文件系统和 FSx Lustre 文件系统是否可用。df -h集群上的输出应类似于以下内容：

```
[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
```

/home文件系统装载了 127.0.0.1，容量非常大。这是您在本教程前面部分创建EFS的文件系统。此处写入的所有文件都将在集群中的所有节点/home上都可用。

/shared文件系统挂载一个私有 IP，容量为 1.2 TB。这是您在本教程前面FSx部分创建的 for Lustre 文件系统。此处写入的所有文件都将在集群中的所有节点/shared上都可用。

与 Slurm 互动

主题

- 列出队列和节点
- 显示职位

列出队列和节点

您可以使用列出队列及其关联的节点sinfo。集群的输出应类似于以下内容：

```
[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 ~]$
```

记下名为的分区demo。它的状态为up，最多有 4 个节点。它与节点组中的compute-1节点相关联。如果您编辑计算节点组并将最大实例数增加到 8，则会读取节点数8并读取节点列表compute-1-

[1-8]。如果您创建了第二个名为 4 个节点test的计算节点组，并将其添加到demo队列中，则这些节点也将显示在节点列表中。

显示职位

您可以使用列出系统上所有处于任何状态的作业squeue。集群的输出应类似于以下内容：

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

当你有 Slurm 任务待处理或正在运行时，请稍后squeue再试运行。

在中运行单节点作业 AWS PCS

要使用 Slurm 运行作业，您需要准备一个指定作业要求的提交脚本，然后使用命令将其提交到队列。sbatch通常，这是在共享目录中完成的，因此登录和计算节点有一个用于访问文件的公共空间。

连接到集群的登录节点，并在其 shell 提示符下运行以下命令。

- 成为默认用户。切换到共享目录。

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

- 使用以下命令创建示例作业脚本：

```
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
```

- 将作业脚本提交给 Slurm 调度器：

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

- 提交作业后，它将以数字形式返回作业 ID。使用该 ID 来检查任务状态。Replace（替换）*job-id* 在以下命令中，返回的数字来自于sbatch。

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

Example

```
squeue --job 1
```

该squeue命令返回的输出类似于以下内容：

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

- 继续检查作业的状态，直到它达到R（正在运行）状态。当squeue没有返回任何东西时，工作就完成了。
- 检查/shared目录的内容。

```
ls -alth /shared
```

命令输出类似于以下内容：

```
-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
```

single.1.err这些文件名为single.1.out、由集群的一个计算节点写入。由于作业是在共享目录 (/shared) 中运行的，因此它们也可以在您的登录节点上使用。这就是您为该集群配置 fo FSx r Lustre 文件系统的原因。

- 检查single.1.out文件内容。

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

输出类似于以下内容：

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

在 Slurm 中运行多节点MPI作业 AWS PCS

这些说明演示了如何使用 Slurm 在中运行消息传递接口 (MPI) 作业。 AWS PCS

在登录节点的 shell 提示符下运行以下命令。

- 成为默认用户。切换到其主目录。

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

- 使用 C 编程语言创建源代码。

```
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
```

- 加载打开MPI模块。

```
module load openmpi
```

- 编译 C 程序。

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

- 编写 Slurm 作业提交脚本。

```
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
```

- 切换到共享目录。

```
cd /shared
```

- 提交作业脚本。

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

- squeue用于监视作业直至其完成。
- 检查以下内容multi.out：

```
cat multi.out
```

输出类似于以下内容。请注意，每个等级都有自己的 IP 地址，因为它运行在不同的节点上。

```
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
```

删除您的 AWS 资源 AWS PCS

完成为本教程创建的集群和节点组后，应删除已创建的资源。

> **Important**
>
> 你需要为你运行的所有资源收取账单费用 AWS 账户

删除您为本教程创建的 AWS PCS资源

- 打开控制AWS PCS台。
- 导航到名为 get-starte d 的集群。

- 导航到 “队列” 部分。
- 选择名为 demo 的队列。
- 选择删除。

> **Important**
> 等到队列被删除后再继续。

- 导航至 “计算节点组” 部分。
- 选择名为 compute-1 的计算节点组。
- 选择删除。
- 选择名为 log in 的计算节点组。
- 选择删除。

> **Important**
> 等到两个计算节点组都被删除后再继续。

- 在用于入门的集群详细信息页面中，选择删除。

> **Important**
> 等到集群被删除后再继续执行后续步骤。

删除您为本教程创建的其他 AWS 资源

- 打开控制IAM台。
  - 选择角色。
  - 选择名为 AWSPCS-getstarted-role 的角色，然后选择删除。
  - 删除角色后，选择策略。
  - 选择名为 AWSPCS-getstarted-policy 的策略，然后选择删除。
- 打开AWS CloudFormation 控制台。
  - 选择名为 getstart ed-lt 的堆栈。
  - 选择删除。

> **Important**
>
> 等待堆栈删除后再继续。

- 打开 Amazon EFS 控制台。
  - 选择文件系统。
  - 选择名为 getstart ed-efs 的文件系统。
  - 选择删除。

> **Important**
>
> 等待文件系统删除后再继续。

- 打开 Amazon FSx 控制台。
  - 选择文件系统。
  - 选择名为 getstart ed-fsx 的文件系统。
  - 选择删除。

> **Important**
>
> 等待文件系统删除后再继续。

- 打开AWS CloudFormation 控制台。
  - 选择名为 getstart ed-sg 的堆栈。
  - 选择删除。
- 打开AWS CloudFormation 控制台。
  - 选择名为 hpc-networking 的堆栈。
  - 选择 Delete（删除）。

AWS PCS集群

集 AWS PCS群由以下组件组成：

- HPC系统调度程序软件的托管实例，例如 Slurm 控制守护程序 ()。`slurmctld`
- 与HPC系统计划程序集成的组件，用于配置和管理 Amazon EC2 实例。
- 与HPC系统调度程序集成的组件，用于向 Amazon CloudWatch 传输日志和指标。

这些组件在由管理的账户中运行 AWS。它们共同管理您的客户账户中的 Amazon EC2 实例。 AWS PCS在您的 Amazon VPC 子网中配置弹性网络接口，以提供从计划程序软件到 Amazon EC2 实例的连接（例如，支持在这些实例上安排批处理任务并使用户能够运行计划程序命令来列出和管理这些任务）。

主题

在 AWS 并行计算服务中创建集群

本主题概述了可用选项，并介绍了在并 AWS 行计算服务 (AWS PCS) 中创建集群时应考虑的事项。如果这是您第一次创建 AWS PCS集群，我们建议您遵循AWS 并行计算服务入门。本教程可以帮助您创建可运行的HPC系统，而无需扩展到所有可能的可用选项和系统架构。

先决条件

- 符合AWS PCS联网要求VPC的现有和子网。在部署用于生产用途的集群之前，我们建议您全面了解VPC和子网的要求。要创建VPC和子网，请参阅VPC为您的 AWS PCS集群创建。
- 具有创建和管理 AWS PCS资源权限的IAM委托人。有关更多信息，请参阅 并 AWS 行计算服务的 Identity and Access 管理。

创建 AWS PCS 集群

您可以使用 AWS Management Console 或 AWS CLI 来创建集群。

AWS Management Console

创建集群

1. 在 https://console.aws.amazon.com/pcs/home#/clusters 上打开 AWS PCS控制台，然后选择创建集群。
2. 在集群设置部分中，输入以下字段：
   - 集群名称-您的集群的名称。名称只能包含字母数字字符（区分大小写）和连字符。它必须以字母字符开头，长度不能超过 40 个字符。该名称在创建集群时 AWS 区域 AWS 账户 使用的名称必须是唯一的。
   - 调度程序-选择调度程序和版本。 AWS PCS目前支持 Slurm 24.05 和 23.11。有关更多信息，请参阅 中的 Slurm 版本 AWS PCS。
   - 控制器大小-选择控制器的大小。这决定了 AWS PCS集群可以管理多少并发任务和计算节点。您只能在创建集群时设置控制器的大小。有关尺码的更多信息，请参阅集群大小为 AWS PCS。
3. 在“网络”部分中，为以下字段选择值：
   - VPC— 选择符合 AWS PCS要求VPC的现有产品。有关更多信息，请参阅 AWS PCS VPC 和子网要求和注意事项。创建集群后，您无法对其进行更改VPC。如果未VPCs列出，则必须先创建一个。
   - 子网-列出了所选VPC子网中的所有可用子网。在不同的可用区中选择两个。每个子网都必须满足子 AWS PCS网要求。有关更多信息，请参阅 AWS PCS VPC 和子网要求和注意事项。我们建议您选择私有子网，以避免将您的调度程序端点暴露给公共 Internet。
   - 安全组-指定 AWS PCS要与其为集群创建的网络接口关联的安全组。您必须至少选择一个允许集群与其计算节点之间通信的安全组。有关更多信息，请参阅 安全组要求和注意事项。
4. （可选）在“加密”下，您可以通过设置以下字段来定义用于加密控制器数据的自定义密钥：
   - KMS密钥 ID — 保留`aws/pcs`为使用PCS创建的KMS密钥。选择现有KMS密钥别名以使用自定义KMS密钥。请注意，用于创建集群的帐户必须具有自定义KMS密钥的`kms:Decrypt`权限。
5. （可选）在 Slurm 配置部分，您可以指定 Slurm 配置选项，这些选项将覆盖通过以下方式设置的默认值： AWS PCS
   - 缩小空闲时间 — 这控制动态配置的计算节点在放置在计算节点上的任务完成或终止后保持活动状态的时间。将其设置为较长的值可以使后续作业更有可能在节点上运行，但可能会导

致成本增加。较小的值可以降低成本，但可能会增加HPC系统在配置节点上而不是在节点上运行作业所花费的时间比例。

- Prolog — 这是指向计算节点组实例上 prolog 脚本目录的完全限定路径。这与 Slurm 中的 Prolog 设置相对应。请注意，这必须是目录，而不是特定可执行文件的路径。
- Epilog — 这是指向计算节点组实例上的 epilog 脚本目录的完全限定路径。这与 Slurm 中的 Epilog 设置相对应。请注意，这必须是目录，而不是特定可执行文件的路径。
- 选择类型参数-这有助于控制 Slurm 使用的资源选择算法。将此值设置为`CR_CPU_Memory`将激活内存感知调度，而将其设置为`CR_CPU`将激活CPU仅限内存的调度。此参数对应于 Slurm 中的SelectTypeParameters设置，其中设置`SelectType`为 by。`select/cons_tres` AWS PCS

6. （可选）在 “标签” 下，将所有标签添加到您的 AWS PCS集群。
7. 选择创建集群。 AWS PCS创建集群`Creating`时会显示状态字段。此过程可能耗时数分钟。

> **Important**
>
> AWS 区域 每个集群只能有 1 个处于同一个`Creating`状态的集群 AWS 账户。 AWS PCS 如果在您尝试创建集群时已经有一个处于`Creating`状态的集群，则返回错误。

AWS CLI

创建集群

1. 使用以下命令创建集群。在运行命令之前，进行以下替换：
   - *`region`*替换为您要 AWS 区域 在其中创建集群的 ID，例如`us-east-1`。
   - 将 *`my-cluster`* 替换为您的集群名称。名称只能包含字母数字字符（区分大小写）和连字符。它必须以字母字符开头，长度不能超过 40 个字符。该名称在创建集群的 AWS 账户 位置 AWS 区域 和创建集群的地方必须是唯一的。
   - *`24.05`*替换为任何支持的 Slurm 版本。

     > **Note**
     >
     > AWS PCS目前支持 Slurm 24.05 和 23.11。

- *SMALL*替换为任何支持的群集大小。这决定了 AWS PCS集群可以管理多少并发任务和计算节点。它只能在创建集群时进行设置。有关尺码的更多信息，请参阅集群大小为 AWS PCS。
- 的值替换为您自己的值。subnetIds我们建议您选择私有子网，以避免将您的调度程序端点暴露给公共 Internet。
- 指定securityGroupIds要与它 AWS PCS为集群创建的网络接口关联的。安全组必须与集群位于同一个VPC安全组中。您必须至少选择一个允许集群与其计算节点之间通信的安全组。有关更多信息，请参阅 安全组要求和注意事项。
- 或者，您可以通过添加选项来微调 Slurm 的行为。--slurm-configration例如，您可以使用将缩小空闲时间设置为 60 分钟（3600 秒）。--slurm configuration scaleDownIdeTime=3600
- 或者，您可以使用自定义KMS密钥来加密控制器的数据--kms-key-id *kms-key*。*kms-key*替换为现有的KMSARN密钥 ID 或别名。请注意，用于创建集群的帐户必须具有自定义KMS密钥的kms:Decrypt权限。

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=24.05 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

2. 配置集群可能需要几分钟。可使用以下命令查询集群的状态。在集群的状态字段变为之前，请勿继续创建队列或计算节点组ACTIVE。

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

> **Important**
>
> AWS 区域 每个集群只能有 1 个处于同一个Creating状态的集群 AWS 账户。 AWS PCS 如果在您尝试创建集群时已经有一个处于Creating状态的集群，则返回错误。

为您的集群推荐的后续步骤

- 添加计算节点组。

- 添加队列。
- 启用日志记录。

删除中的集群 AWS PCS

本主题概述了如何删除集AWSPCS群。

删除 AWS PCS集群时的注意事项

- 必须先删除与集群关联的所有队列，然后才能删除集群。有关更多信息，请参阅 删除中的队列 AWS PCS。
- 必须先删除与集群关联的所有计算节点组，然后才能删除集群。有关更多信息，请参阅 删除中的计算节点组 AWS PCS。

删除集群

您可以使用 AWS Management Console 或 AWS CLI 来删除集群。

AWS Management Console

删除集群

1. 打开控制AWS PCS台。
2. 选择要删除的集群。
3. 选择删除。
4. 将显示集群状态字段Deleting。可能需要几分钟的时间才能完成。

AWS CLI

删除集群

1. 使用以下命令删除集群，并使用以下替换命令：
   - Replace（替换）*region-code* AWS 区域 你的集群就在里面。
   - Replace（替换）*my-cluster* 使用您的集群的名称或 ID。

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

2. 删除集群可能需要几分钟。您可以使用以下命令检查集群的状态。

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

集群大小为 AWS PCS

AWS PCS提供高度可用且安全的群集，同时自动执行修补、节点配置和更新等关键任务。

创建集群时，您可以根据两个因素为其选择大小：

- 它将管理的计算节点数量
- 您预计将在集群上运行的活跃作业和排队作业的数量

> **Important**
>
> 创建集群后，您无法更改集群大小。如果需要更改大小，则必须创建一个新集群。

Slurm 集群大小	托管的实例数量	活跃和排队的作业数量
小型	最多 32	最多 256
中	最多 512	最高 8192
大型	直到 2048	最高 16384

示例

- 如果您的集群将有多达 24 个托管实例并运行多达 100 个作业，请选择 Sm all。
- 如果您的集群将有多达 24 个托管实例并运行多达 1000 个作业，请选择 “中”。
- 如果您的集群将有多达 1000 个托管实例并运行多达 100 个作业，请选择大型。
- 如果您的集群将有多达 1000 个托管实例并运行多达 10,000 个作业，请选择大型。

在中使用集群密钥 AWS PCS

作为创建集群的一部分， AWS PCS创建连接到集群上的作业调度器所需的集群密钥。您还可以创建 AWS PCS计算节点组，这些节点组定义了为响应扩展事件而启动的实例集。 AWS PCS使用集群密钥配置由这些计算节点组启动的实例，以便它们可以连接到作业调度器。在某些情况下，您可能需要手动配置 Slurm 客户端。示例包括构建永久登录节点或设置具有作业管理功能的工作流管理器。

AWS PCS将集群密钥存储为托管密钥，前缀为pcs!中 AWS Secrets Manager。密钥的费用包含在使用费用中 AWS PCS。

> **Warning**
>
> 请勿修改您的集群密钥。 AWS PCS如果您修改集群密钥，将无法与您的集群通信。 AWS PCS不支持集群密钥的轮换。如果您需要修改集群密钥，则必须创建一个新集群。

目录

用于 AWS Secrets Manager 查找集群密钥

AWS Management Console

1. 导航到 Secrets Manager 控制台。
2. 选择 “密钥”，然后搜索前pcs!缀。

> **Note**
>
> 集 AWS PCS群密钥的名称形式为 AWS PCS集群 ID，pcs!slurm-secret-*cluster-id*其中*cluster-id*。

AWS CLI

每个 AWS PCS集群密钥也都标有`aws:pcs:`*`cluster-id`*。您可以使用以下命令获取集群的密钥 ID。在运行命令之前，请进行以下替换：

- *`region`*替换 AWS 区域 为可在其中创建集群，例如`us-east-1`。
- *`cluster-id`*替换为要查找集 AWS PCS群密钥的集群 ID。

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

用于 AWS PCS查找集群密钥

您可以使用 AWS CLI 来查找集ARN AWS PCS群密钥。输入以下命令，进行以下替换：

- *`region`*替换 AWS 区域 为可在其中创建集群，例如`us-east-1`。
- *`my-cluster`*替换为您的集群的名称或标识符。

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

以下示例输出来自该`get-cluster`命令。你可以`secretVersion`一起使用`secretArn`和来获得秘密。

```
{
    "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": "24.05"
        },
        "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"
            }
        ]
    }
}
```

获取 Slurm 集群的秘密

你可以使用 Secrets Manager 获取 Slurm 集群密钥的当前 base64 编码版本。以下示例使用了。 AWS CLI在运行命令之前，请进行以下替换。

- *region*替换 AWS 区域 为可在其中创建集群，例如us-east-1。
- *secret-arn*替换为secretArn来自 AWS PCS群集的。

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

有关如何使用 Slurm 集群密钥的信息，请参阅。[使用独立实例作为 AWS PCS登录节点]

权限

你使用IAM委托人获取 Slurm 集群密钥。IAM校长必须拥有读取秘密的权限。有关更多信息，请参阅AWS Identity and Access Management 用户指南中的角色术语和概念。

以下示例IAM策略允许访问示例集群密钥。

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

AWS PCS计算节点组

AWS PCS计算节点组是节点的逻辑集合（Amazon EC2 实例）。这些节点可用于运行计算作业，以及提供基于外壳的交互式系统访问权限。HPC计算节点组由创建节点的规则组成，包括要使用的 Amazon EC2 实例类型、要运行的实例数量、使用竞价型实例还是按需实例、要使用哪些子网和安全组，以及如何在每个实例启动时对其进行配置。更新这些规则后，会 AWS PCS更新与计算节点组关联的资源以使其匹配。

主题

在中创建计算节点组 AWS PCS

本主题概述了可用选项，并介绍了在并 AWS 行计算服务 (AWS PCS) 中创建计算节点组时应考虑的事项。如果这是您第一次在中创建计算节点组 AWS PCS，我们建议您按照中的教程进行操作AWS 并行计算服务入门。本教程可以帮助您创建可运行的HPC系统，而无需扩展到所有可能的可用选项和系统架构。

先决条件

- 足够的服务配额可以在您的中启动所需数量的EC2实例 AWS 区域。您可以使用AWS Management Console来检查和请求增加服务配额。
- 满足 AWS PCS网络要求的现有子网VPC和子网。我们建议您在部署用于生产的集群之前，充分了解这些要求。有关更多信息，请参阅 AWS PCS VPC 和子网要求和注意事项。您也可以使用 CloudFormation 模板来创建VPC和子网。 AWS 提供了 CloudFormation 模板的HPC配方。有关更多信息，请参阅 aws-hpc-recipes上的 GitHub。
- IAM实例配置文件，有权调用 AWS PCS`RegisterComputeNodeGroupInstance`API操作并访问您的节点组实例所需的任何其他 AWS 资源。有关更多信息，请参阅 AWS 并行计算服务的 IAM 实例配置文件。
- 您的节点组实例的启动模板。有关更多信息，请参阅 将 Amazon EC2 启动模板与 AWS PCS。
- 要创建使用 Amazon EC2 Spot 实例的计算节点组，您必须拥有AWSServiceRoleForEC2Spot服务相关角色。 AWS 账户有关更多信息，请参阅 适用于 AWS PCS 的 Amazon EC2 Spot 角色。

在中创建计算节点组 AWS PCS

您可以使用 AWS Management Console 或创建计算节点组 AWS CLI。

AWS Management Console

使用控制台创建计算节点组

1. 打开控制AWS PCS台。
2. 选择要在其中创建计算节点组的集群。导航到 “计算节点组”，然后选择 “创建”。
3. 在计算节点组设置部分，为您的节点组提供一个名称。名称只能包含区分大小写的字母数字字符和连字符。它必须以字母字符开头，长度不能超过 25 个字符。该名称在集群中必须是唯一的。
4. 在 “计算配置” 下，输入或选择以下值：

   a. EC2启动模板-选择用于此节点组的自定义启动模板。启动模板可用于自定义网络设置，例如子网、安全组、监控配置和实例级存储。如果您尚未准备好启动模板，请参阅将 Amazon EC2 启动模板与 AWS PCS以了解如何创建启动模板。

   > **Important**
   >
   > AWS PCS为每个计算节点组创建托管启动模板。这些都被命名`pcs-identifier-do-not-delete`了。创建或更新计算节点组时请勿选择这些，否则节点组将无法正常运行。

   b. EC2启动模板版本-您必须选择自定义启动模板的版本。如果稍后更改版本，则必须更新计算节点组以检测启动模板中的更改。有关更多信息，请参阅 更新 AWS PCS计算节点组。

   c. AMIID — 如果您的启动模板不包含 AMI ID，或者您想覆盖启动模板中的值，请在此处提供一个 AMI ID。请注意，AMI用于节点组的必须与兼容 AWS PCS。您也可以选择AMI提供的样本 AWS。有关此主题的更多信息，请参阅Amazon 机器映像 (AMIs) 适用于 AWS PCS。

   d. IAM实例配置文件-为节点组选择实例配置文件。实例配置文件授予实例安全访问 AWS 资源和服务的权限。如果您还没有准备好，请参阅AWS 并行计算服务的 IAM 实例配置文件以了解如何创建一个。

   e. 子网-在 AWS PCS集群部署VPC位置中选择一个或多个子网。如果您选择多个子网，则节点之间将无法进行EFA通信，并且不同子网中的节点之间的通信可能会增加延迟。确保您在此处指定的子网与您在EC2启动模板中定义的任何子网相匹配。

f. 实例-选择一个或多个实例类型来满足节点组中的扩展请求。所有实例类型都必须具有相同的处理器架构（x86_64 或 arm64）和数量。vCPUs如果实例有GPUs，则所有实例类型必须具有相同数量的GPUs。

g. 扩展配置-指定节点组的最小和最大实例数。您可以定义静态配置（其中有固定数量的节点在运行），也可以定义动态配置，其中最多可以运行最大数量的节点。对于静态配置，请将最小值和最大值设置为相同的、大于零的数字。对于动态配置，请将最小实例数设置为零，将最大实例数设置为大于零的数字。 AWS PCS不支持混合使用静态和动态实例的计算节点组。

5. （可选）在 “其他设置” 下，指定以下内容：

a. 购买选项-在 Spot 实例和按需实例之间进行选择。

b. 分配策略 — 如果您选择了竞价购买选项，则可以指定在启动节点组中的实例时如何选择竞价容量池。有关更多信息，请参阅 Amazon 弹性计算云用户指南中的竞价型实例分配策略。如果您选择了按需购买选项，则此选项无效。

6. （可选）在 Slurm 自定义设置部分，请提供以下值：

a. 权重-此值用于设置组中节点的优先级，以便进行调度。权重较低的节点具有更高的优先级，并且单位是任意的。有关更多信息，请参阅《中的重量》 Slurm 文档中）。

b. 实际内存-此值设置节点组中节点上实际内存的大小（以 GB 为单位）。它本应与集群中的`CR_CPU_Memory`选项一起使用 Slurm 配置在 AWS PCS。有关更多信息，请参阅RealMemory中的 Slurm 文档中）。

7. （可选）在 “标签” 下，将所有标签添加到您的计算节点组。

8. 选择创建计算节点组。置备节点组**`Creating`**时 AWS PCS会显示状态字段。这个过程可能需要几分钟。

建议采取下一步行动

- 将您的节点组添加到中的队列中 AWS PCS，使其能够处理作业。

AWS CLI

要创建您的计算节点组，请使用以下命令 AWS CLI

使用以下命令创建队列。在运行命令之前，进行以下替换：

1. Replace（替换）*`region`*带有 AWS 区域 用于创建集群的 ID，例如`us-east-1`。

2. Replace（替换）*my-cluster* 使用您的集群clusterId的名称或。
3. Replace（替换）*my-node-group*使用您的计算节点组的名称。名称只能包含字母数字字符（区分大小写）和连字符。它必须以字母字符开头，长度不能超过 25 个字符。该名称在集群中必须是唯一的。
4. Replace（替换）*subnet-ExampleID1* 使用集群IDsVPC中的一个或多个子网。
5. Replace（替换）*lt-ExampleID1* 使用您的自定义启动模板的 ID。如果您还没有准备好，请参阅将 Amazon EC2 启动模板与 AWS PCS以了解如何创建一个。

   > **Important**
   >
   > AWS PCS为每个计算节点组创建托管启动模板。这些都被命名pcs-*identifier*-do-not-delete了。创建或更新计算节点组时请勿选择这些，否则节点组将无法正常运行。

6. Replace（替换）*launch-template-version* 使用特定的启动模板版本。 AWS PCS将您的节点组与该特定版本的启动模板相关联。
7. Replace（替换）*arn:InstanceProfile*使用您的IAM实例配置文件中的。ARN如果您还没有准备好，请参阅将 Amazon EC2 启动模板与 AWS PCS获取指导。
8. Replace（替换）*min-instances* 以及 *max-instances* 使用整数值。您可以定义静态配置（其中有固定数量的节点在运行），也可以定义动态配置，其中最多可以运行最大数量的节点。对于静态配置，请将最小值和最大值设置为相同的、大于零的数字。对于动态配置，请将最小实例数设置为零，将最大实例数设置为大于零的数字。 AWS PCS不支持混合使用静态和动态实例的计算节点组。
9. Replace（替换）*t3.large* 使用另一种实例类型。您可以通过指定instanceType设置列表来添加更多实例类型。例如，*--instance-configs instanceType=c6i.16xlarge,instanceType=c6a.16xlarge*。 所有实例类型都必须具有相同的处理器架构（x86_64 或 arm64）和数量。vCPUs如果实例有GPUs，则所有实例类型必须具有相同数量的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
```

您可以将几个可选的配置设置添加到create-compute-node-group命令中。

- 您可以指定您的自定义启动模板--amiId是否不包含对的引用AMI，或者您是否希望覆盖该值。请注意，AMI用于节点组的必须与兼容 AWS PCS。您也可以选择AMI提供的样本 AWS。有关此主题的更多信息，请参阅Amazon 机器映像 (AMIs) 适用于 AWS PCS。
- 您可以使用在按需 (ONDEMAND) 和 Spot (SPOT) 实例之间进行选择--purchase-option。按需是默认设置。如果您选择竞价型实例，则还可以使用--allocation-strategy来定义当竞价型容量池启动节点组中的实例时如何 AWS PCS选择竞价型容量池。有关更多信息，请参阅 Amazon 弹性计算云用户指南中的竞价型实例分配策略。
- 可以提供 Slurm 使用为节点组中的节点配置选项--slurm-configuration。您可以设置权重（调度优先级）和实际内存。权重较低的节点具有更高的优先级，并且单位是任意的。有关更多信息，请参阅《中的重量》 Slurm 文档中）。实际内存设置节点组中节点上实际内存的大小（以 GB 为单位）。它本应与您的集群 AWS PCS中的CR_CPU_Memory选项结合使用 Slurm 配置。有关更多信息，请参阅RealMemory中的 Slurm 文档中）。

> **Important**
>
> 创建计算节点组可能需要几分钟。

您可以使用以下命令查询节点组的状态。在节点组的状态达到之前，您将无法将其与队列关联ACTIVE。

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

更新 AWS PCS计算节点组

本主题概述了可用选项，并介绍了更新AWSPCS计算节点组时应考虑的事项。

更新AWSPCS计算节点组的选项

通过更新AWSPCS计算节点组，您可以更改由AWSPCS启动的实例的属性以及启动这些实例的规则。例如，您可以将节点组实例替换AMI为另一个安装了不同软件的实例。或者，您可以更新安全组以更改入站或出站网络连接。您还可以更改扩展配置，甚至更改竞价型实例的首选购买选项。

以下节点组设置在创建后无法更改：

- 名称
- 实例

更新 AWS PCS计算节点组时的注意事项

计算节点组定义了用于处理作业、提供交互式外壳访问权限和其他任务的EC2实例。它们通常与一个或多个 AWS PCS队列相关联。在更新计算节点组以更改其行为（或其节点的行为）时，请考虑以下几点：

- 当计算节点组状态从 “更新” 变为 “活动” 时，对计算节点组属性的更改就会生效。使用更新的属性启动新实例。
- 不影响特定节点配置的更新不会影响正在运行的节点。例如，添加子网和更改分配策略。
- 如果您更新计算节点组的启动模板，则必须更新计算节点组才能使用新版本。
- 要在计算节点组的节点中添加或移除安全组，请编辑其启动模板并更新计算节点组。使用更新的安全组集启动新实例。
- 如果您直接编辑计算节点组使用的安全组，则该安全组会立即对正在运行的实例和 future 实例生效。
- 如果您在计算节点组使用的IAM实例配置文件中添加或删除权限，它将立即对正在运行的实例和将来的实例生效。
- 要更改计算节点组的实例所AMI使用的实例，请将计算节点组（或其启动模板）更新为使用新的，AMI然后等待 AWS PCS替换实例。
- AWS PCS节点组更新操作后替换节点组中的现有实例。如果某个节点上正在运行作业，则允许这些任务在 AWS PCS替换该节点之前完成。交互式用户进程（例如在登录节点实例上）终止。当将实例AWS PCS标记为替换`Active`时，节点组状态会恢复为，但实际替换发生在实例处于空闲状态时。
- 如果您减少计算节点组中允许的最大实例数，则会从 Slurm 中 AWS PCS移除节点以达到新的最大值。 AWS PCS终止与已移除的 Slurm 节点关联的正在运行的实例。已移除的节点上正在运行的作业失败并返回其队列。

- AWS PCS为每个计算节点组创建托管启动模板。他们被命名pcs-*identifier*-do-not-delete了。创建或更新计算节点组时请勿选择它们，否则节点组将无法正常运行。
- 如果您更新计算节点组以使用 Spot 作为其购买选项，则您的账户中必须具有AWSServiceRoleForEC2Spot服务相关角色。有关更多信息，请参阅 适用于 AWS PCS 的 Amazon EC2 Spot 角色。

更新AWSPCS计算节点组

您可以使用AWS管理控制台或更新节点组AWSCLI。

AWS Management Console

更新计算节点组

1. 在以下位置打开AWSPCS控制台 https://console.aws.amazon.com/pcs/home#/clusters
2. 选择要在其中更新计算节点组的集群。
3. 导航到计算节点组，转到要更新的节点组，然后选择编辑。
4. 在 “计算配置”、“其他设置” 和 “Slurm自定义设置” 部分中，更新除以下值之外的所有值：
   - 实例-您无法更改计算节点组中的实例。
5. 选择更新。应用更改时，“状态” 字段将显示 “正在更新”。

> **Important**
> 计算节点组更新可能需要几分钟时间。

AWS CLI

更新计算节点组

1. 使用以下命令更新您的计算节点组。在运行命令之前，进行以下替换：
   a. Replace（替换）*region-code* 以及您要在其中创建集群的AWS区域。
   b. Replace（替换）*my-node-group* 使用您的计算节点组computeNodeGroupId的名称或。

c. Replace（替换）*my-cluster* 使用您的集群clusterId的名称或。

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

2. 更新除之外的所有节点组参数--instance-configs。例如，要设置新 AMI ID，请传递 wher --amiId my-custom-ami-id e *my-custom-ami-id* 已由您选择AMI的取而代之。

> **Important**
>
> 更新计算节点组可能需要几分钟。

您可以使用以下命令查询节点组的状态。

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

删除中的计算节点组 AWS PCS

本主题概述了可用选项，并介绍了在中删除计算节点组时应考虑的事项 AWS PCS。

删除计算节点组时的注意事项

计算节点组定义用于处理作业、提供交互式外壳访问权限和其他任务的EC2实例。它们通常与一个或多个 AWS PCS队列相关联。在删除计算节点组之前，请考虑以下事项：

- 计算节点组启动的所有EC2实例都将被终止。这将取消在这些实例上运行的作业，并终止正在运行的交互式进程。
- 必须先取消计算节点组与所有队列的关联，然后才能将其删除。有关更多信息，请参阅 更新队 AWS PCS列。

删除计算节点组

您可以使用 AWS Management Console 或 AWS CLI 删除计算节点组。

AWS Management Console

删除计算节点组

1. 打开控制AWS PCS台。
2. 选择计算节点组的集群。
3. 导航到计算节点组，然后选择要删除的计算节点组。
4. 选择删除。
5. 将显示 “状态” 字段`Deleting`。可能需要几分钟的时间才能完成。

> **Note**
>
> 您可以使用调度程序原生的命令来确认计算节点组已删除。例如，对于 Slur `squeue m` 使用`sinfo`或。

AWS CLI

删除计算节点组

- 使用以下命令删除具有以下替换内容的计算节点组：
  - Replace（替换）*`region-code`* AWS 区域 你的集群就在里面。
  - Replace（替换）*`my-node-group`* 使用您的计算节点组的名称或 ID。
  - Replace（替换）*`my-cluster`* 使用您的集群的名称或 ID。

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

删除计算节点组可能需要几分钟。

> **Note**
>
> 您可以使用调度程序原生的命令来确认计算节点组已删除。例如，对于 Slur squeue m 使用sinfo或。

在中查找计算节点组实例 AWS PCS

每个 AWS PCS计算节点组都可以启动具有共享配置的EC2实例。您可以使用EC2标签在 AWS Management Console 或的计算节点组中查找实例 AWS CLI。

AWS Management Console

查找您的计算节点组实例

1. 打开控制AWS PCS台。
2. 选择 集群。
3. 选择计算节点组。
4. 查找您创建的登录节点组的 ID。
5. 导航到EC2控制台并选择实例。
6. 搜索带有以下标签的实例。Replace（替换）*node-group-id* 使用您的计算节点组的 ID（不是名称）。

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

7. （可选）您可以在搜索字段中更改实例状态的值，以查找正在配置或最近终止的实例。
8. 在已标记的实例列表中查找每个实例的实例 ID 和 IP 地址。

AWS CLI

要查找您的节点组实例，请使用以下命令。在运行命令之前，请进行以下替换：

- *region-code*替换为您的 AWS 区域 集群的。例如：us-east-1
- *node-group-id*替换为计算节点组的 ID（不是名称）。
- running替换为其他实例状态（例如pending或）terminated以查找处于其他状态的EC2实例。

```
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}
```

该命令返回的输出类似于下方内容。null如果实例PublicIP位于私有子网中，则值为。

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

> **Note**
>
> 如果您希望describe-instances返回大量实例，则必须对多个页面使用选项。有关更多信息，请参阅DescribeInstances《Amazon 弹性计算云API参考》。

将 Amazon EC2 启动模板与 AWS PCS

在 Amazon 中EC2，启动模板可以存储一组首选项，这样您就不必在启动实例时单独指定它们。 AWS PCS整合了启动模板作为配置计算节点组的灵活方式。创建节点组时，您需要提供启动模板。 AWS PCS从中创建包含转换的派生启动模板，以帮助确保其与服务配合使用。

了解编写自定义启动模板时有哪些选项和注意事项，可以帮助您编写一个供您使用的模板 AWS PCS。有关启动模板的更多信息，请参阅 Amazon EC2 用户指南中的通过启动模板启动实例。

主题

中的启动模板概述 AWS PCS

您可以在EC2启动模板中包含 30 多个参数，控制实例配置方式的许多方面。大多数都与完全兼容 AWS PCS，但也有一些例外。

EC2Launch 模板的以下参数将被忽略， AWS PCS因为这些属性必须由服务直接管理：

- 实例类型/指定实例类型属性 (`InstanceRequirements`)- AWS PCS 不支持基于属性的实例选择。
- 实例类型 (`InstanceType`)-在创建节点组时指定实例类型。
- 高级详细信息/ IAM 实例配置文件 (`IamInstanceProfile`)-您在创建或更新节点组时提供此信息。
- 高级详细信息/禁用API终止 (`DisableApiTermination`) — AWS PCS 必须控制其启动的节点组实例的生命周期。
- 高级详细信息/禁用 API stop (`DisableApiStop`) — AWS PCS 必须控制其启动的节点组实例的生命周期。
- 高级详情/停止 — Hibernate 行为 (`HibernationOptions`) — AWS PCS 不支持实例休眠。
- 高级详情/弹性 GPU (`ElasticGpuSpecifications`) — 亚马逊 Elastic Graphics 于 2024 年 1 月 8 日停产。

- 高级详细信息/弹性推理 (ElasticInferenceAccelerators) — Amazon Elastic Inference 不再向新客户开放。
- AAdvanceddetails/Specify CPU options/Threads每核 (ThreadsPerCore) — AWS PCS 将每个内核的线程数设置为 1。

这些参数有特殊要求，需要支持与 AWS PCS以下各项兼容：

- 用户数据 (UserData)-必须进行多部分编码。请参阅 使用 Amazon EC2 用户数据。
- 应用程序和操作系统映像 (ImageId)-您可以将其包括在内。但是，如果您在创建或更新节点组时指定 AMI ID，它将覆盖启动模板中的值。AMI您提供的必须与兼容 AWS PCS。有关更多信息，请参阅“Amazon 机器映像 (AMIs) 适用于 AWS PCS。
- 网络设置/防火墙（安全组**SecurityGroups**）()-无法在 AWS PCS启动模板中设置安全组名称列表。除非您在启动模板中定义网络接口，否则您可以设置安全组列表 IDs (SecurityGroupIds)。然后，必须IDs为每个接口指定安全组。有关更多信息，请参阅 中的安全组 AWS PCS。
- 网络设置/高级网络配置 (NetworkInterfaces) — 如果您使用带有单个网卡的EC2实例，并且不需要任何专门的网络配置，则 AWS PCS可以为您配置实例联网。要配置多个网卡或在您的实例上启用弹性结构适配器，请使用NetworkInterfaces。每个网络接口IDs下都必须有一个安全组列表Groups。有关更多信息，请参阅 里面有多个网络接口 AWS PCS。
- 高级详细信息/容量预留 (CapacityReservationSpecification)-可以设置，但在使用CapacityReservationId时不能引用具体内容。 AWS PCS但是，您可以引用容量预留组，该组包含一个或多个容量预留。有关更多信息，请参阅 中的容量预留 AWS PCS。

创建基本的启动模板

您可以使用 AWS Management Console 或创建启动模板 AWS CLI。

AWS Management Console

创建启动模板

1. 打开 Amazon EC2 控制台，然后选择 “启动模板”。
2. 选择Create launch template（创建启动模板）。
3. 在 Launch 模板名称和描述下，为 La unch 模板名称输入一个唯一且独特的名称
4. 在 “密钥对名称” 的 “密钥对（登录）” 下，选择将用于登录由管理的EC2实例的SSH密钥对 AWS PCS。您可以自由选择，但我们建议您这样做。

5. 在网络设置下，然后选择防火墙（安全组），选择要连接到网络接口的安全组。启动模板中的所有安全组都必须来自您的 AWS PCS集群VPC。至少选择：

   - 允许与 AWS PCS集群通信的安全组
   - 允许由启动的EC2实例之间进行通信的安全组 AWS PCS
   - （可选）允许对交互式实例进行入站SSH访问的安全组
   - （可选）允许计算节点与 Internet 建立传出连接的安全组
   - （可选）允许访问网络资源（例如共享文件系统或数据库服务器）的安全组。

6. 您可以在亚马逊EC2控制台的 “启动模板” 下方访问您的新启动模板 ID。启动模板 ID 将包含表单`lt-0123456789abcdef01`。

下一步推荐

- 使用新的启动模板创建或更新 AWS PCS计算节点组。

AWS CLI

创建启动模板

使用以下命令创建启动模板。

- 在运行命令之前，进行以下替换：

  a. Replace（替换）*`region-code`* 以及你 AWS 区域 在哪里工作 AWS PCS

  b. Replace（替换）*`my-launch-template-name`* 用你的模板的名字。它必须是 AWS 区域 您正在使用的唯一的。 AWS 账户

  c. Replace（替换）*`my-ssh-key-name`* 使用您的首选SSH密钥的名称。

  d. Replace（替换）*`sg-ExampleID1`* 以及 *`sg-ExampleID2`* 使用安全组IDs，允许您的EC2实例与计划程序之间进行通信以及EC2实例之间的通信。如果您只有一个安全组可以启用所有这些流量，则可以删除`sg-ExampleID2`其前面的逗号字符。您还可以添加更多安全组IDs。您在启动模板中包含的所有安全组都必须来自您的 AWS PCS集群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 将输出类似于以下内容的文本。启动模板 ID 可在中找到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"
    }
}
```

下一步推荐

- 使用新的启动模板创建或更新 AWS PCS计算节点组。

使用 Amazon EC2 用户数据

您可以在实例启动时cloud-init运行的启动模板中提供EC2用户数据。内容类型的用户数据块在实例向注册之前cloud-config运行 AWS PCSAPI，而内容类型的用户数据块在注册完成后但在 Slurm 守护程序启动之前text/x-shellscript运行。有关内容类型的更多信息，请参阅 cloud-init 文档。

我们的用户数据可以执行常见的配置场景，包括但不限于以下情况：

- 包括用户或群组
- 安装软件包
- 创建分区和文件系统
- 挂载网络文件系统

启动模板中的用户数据必须采用MIME多部分存档格式。这是因为您的用户数据与在节点组中配置节点所需的其他 AWS PCS用户数据合并。您可以将多个用户数据块合并到一个由MIME多个部分组成的文件中。

一个由MIME多部分组成的文件由以下部分组成：

- 内容类型和段边界声明：Content-Type: multipart/mixed; boundary="==BOUNDARY=="
- MIME版本声明：MIME-Version: 1.0

- 一个或多个用户数据块，其包含以下组成部分：
  - 开口边界，表示用户数据块的开头：`--==BOUNDARY==`必须将此边界之前的行留空。
  - 区块的内容类型声明：`Content-Type: text/cloud-config; charset="us-ascii"`或`Content-Type: text/x-shellscript; charset="us-ascii"`。必须将内容类型声明之后的行留空。
  - 用户数据的内容，例如，Shell 命令或 `cloud-config` 指令的列表。
- 表示MIME多部分文件结束的闭合边界:`--==BOUNDARY==--`. 必须将此闭合边界之前的行留空。

> **Note**
>
> 如果您在 Amazon EC2 控制台的启动模板中添加用户数据，则可以将其粘贴为纯文本。或者，您可以从文件上传它。如果使用 AWS CLI 或 AWS SDK，则必须先对用户数据进行 base64 编码，并在调用时将该字符串作为`UserData`参数值提交 CreateLaunchTemplate，如本JSON文件所示。

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

示例

- 示例：从软件包存储库安装软件
- 示例：从 S3 存储桶运行脚本
- 示例：设置全局环境变量
- 将网络文件系统与 AWS PCS
- 示例：使用EFS文件系统作为共享主目录

示例： AWS PCS从软件包存储库安装软件

在启动模板"userData"中提供此脚本作为的值。有关更多信息，请参阅 使用 Amazon EC2 用户数据。

此脚本使用 cloud-config 在启动时在节点组实例上安装软件包。有关更多信息，请参阅 cloud-in it 文档中的用户数据格式。此示例安装curl和llvm。

> **Note**
>
> 您的实例必须能够连接到其配置的软件包存储库。

```
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==--
```

示例： AWS PCS从 S3 存储桶运行其他脚本

在启动模板"userData"中提供此脚本作为的值。有关更多信息，请参阅 使用 Amazon EC2 用户数据。

以下用户数据脚本使用 cloud-config 从 S3 存储桶导入脚本，并在启动时在节点组实例上运行该脚本。有关更多信息，请参阅 cloud-in it 文档中的用户数据格式。

用您自己的详细信息替换以下值：

- *amzn-s3-demo-bucket* — 您的账户可以读取的 S3 存储桶的名称。
- *object-key* — 要导入的脚本的 S3 对象密钥。这包括脚本的名称及其在存储桶文件夹结构中的位置。例如，scripts/script.sh。有关更多信息，请参阅《亚马逊简单存储服务用户指南》中的使用文件夹在 Amazon S3 控制台中组织对象。

- *shell* — 用于运行脚本的 Linux 外壳，例如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==--
```

节点组的IAM实例配置文件必须有权访问存储桶。以下IAM策略是上述用户数据脚本中存储桶的示例。

```
{
    "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/*"
            ]
        }
    ]
}
```

示例：为设置全局环境变量 AWS PCS

在启动模板"userData"中提供此脚本作为的值。有关更多信息，请参阅 使用 Amazon EC2 用户数据。

以下示例用于/etc/profile.d在节点组实例上设置全局变量。

```
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==--
```

示例：使用EFS文件系统作为共享的主目录 AWS PCS

在启动模板"userData"中提供此脚本作为的值。有关更多信息，请参阅 使用 Amazon EC2 用户数据。

此示例扩展了EFS装入的示例将网络文件系统与 AWS PCS，以实现共享的主目录。/home 的内容会在挂载EFS文件系统之前进行备份。然后在装载完成后将内容快速复制到共享存储器上。

用您自己的详细信息替换此脚本中的以下值：

- *`/mount-point-directory`* — 要在实例上挂载EFS文件系统的路径。
- *`filesystem-id`* — 文件系统的EFS文件系统 ID。

```
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==--
```

示例：启用无密码 SSH

您可以在共享主目录示例的基础上使用SSH密钥实现集群实例之间的SSH连接。对于使用共享主文件系统的每个用户，请运行类似于以下内容的脚本：

```
#!/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**
>
> 这些实例必须使用允许在群集节点之间SSH建立连接的安全组。

中的容量预留 AWS PCS

您可以使用按需容量预留或EC2EC2容量块在特定可用区域中预留 Amazon 容量，并在特定期限内预留 Amazon 容量，以确保在需要时有必要的计算容量可用。

> **Note**
>
> AWS PCS支持按需容量预留 (ODCR)，但目前不支持 ML 的容量块。

ODCRs与一起使用 AWS PCS

您可以选择预留 AWS PCS实例的使用方式。如果您创建了打开的ODCR，则您的账户中由 AWS PCS或其他流程启动的任何匹配实例都将计入预留中。对于定向ODCR，只有使用特定预留 ID 启动的实例才会计入预留。对于时间敏感型工作负载，定向ODCRs更为常见。

您可以将 AWS PCS计算节点组配置为使用目标，ODCR方法是将其添加到启动模板中。以下是执行此操作的步骤：

1. 创建有针对性的按需容量预留 (ODCR)。
2. 将ODCR添加到容量预留组。
3. 将容量预留组与启动模板关联。
4. 创建或更新 AWS PCS计算节点组以使用启动模板。

示例：预留并使用具有目标的 hpc6a.48xlarge 实例 ODCR

此示例命令为 32 个 hpc6a.48 ODCR xlarge 实例创建了一个目标。要在置放群组中启动预留实例，请`--placement-group-arn`向命令中添加。您可以使用`--end-date`和定义停止日期`--end-date-type`，否则预留将一直持续到手动终止。

```
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
```

此命令的结果将是新ARN的ODCR。要ODCR与一起使用 AWS PCS，必须将其添加到容量预留组中。这是因为 AWS PCS不支持个人ODCRs。有关更多信息，请参阅 Amazon 弹性计算云用户指南中的容量预留组。

以下是将添加到名ODCR为的容量预留组的方法`EXAMPLE-CR-GROUP`。

```
aws resource-groups group-resources --group EXAMPLE-CR-GROUP \
    --resource-arns arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1
```

ODCR创建容量预留组并将其添加到容量预留组后，现在可以通过将其添加到启动模板来将其连接到 AWS PCS计算节点组。以下是引用容量预留组的启动模板示例。

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

```
}
```

最后，创建或更新 AWS PCS计算节点组以使用 hpc6a.48xlarge 实例，并使用其容量预留组中引用的启动模板。ODCR对于静态节点组，将最小和最大实例数设置为预留的大小 (32)。对于动态节点组，将最小实例数设置为 0，将最大实例数设置为预留大小。

此示例是为一个计算节点组配置ODCR的单个节点的简单实现。但是， AWS PCS支持许多其他设计。例如，您可以将大型ODCR或容量预留组细分为多个计算节点组。或者ODCRs，您可以使用其他AWS帐户创建并与您共享的帐户。关键约束条件是必须ODCRs始终包含在容量预留组中。

有关更多信息，请参阅 Amazon 弹性计算云用户指南中的机器学习按需容量预留和容量块。

有用的启动模板参数

本节介绍一些可能广泛使用的启动模板参数 AWS PCS。

开启详细 CloudWatch监控

您可以使用启动模板参数在较短的时间间隔内启用 CloudWatch 指标收集。

AWS Management Console

在用于创建或编辑启动模板的控制台页面上，可以在高级详细信息部分下找到此选项。将 “详细 CloudWatch 监控” 设置为 “启用”。

YAML

```
Monitoring:
    Enabled: True
```

JSON

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

有关更多信息，请参阅适用于 Linux 实例的 Amazon 弹性计算云用户指南中的启用或关闭实例的详细监控。

实例元数据服务版本 2 (IMDSv2)

将 IMDS v2 与EC2实例配合使用可显著增强安全性，并有助于降低与在 AWS 环境中访问实例元数据相关的潜在风险。

AWS Management Console

在用于创建或编辑启动模板的控制台页面上，可以在高级详细信息部分下找到此选项。将可访问的元数据设置为已启用，将元数据版本设置为仅限 V2（需要令牌），将元数据响应跳跃限制设置为 4。

YAML

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

JSON

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

.

AWS PCS队列

AWS PCS队列是对调度器对工作队列的本机实现的轻量级抽象。就 Slurm 而言， AWS PCS队列等同于 Slurm 分区。

用户将作业提交到他们所在的队列，直到可以安排作业在一个或多个计算节点组提供的节点上运行。一个 AWS PCS集群可以有多个任务队列。例如，您可以创建一个使用亚马逊EC2按需实例处理高优先级任务的队列和另一个使用亚马逊EC2竞价型实例处理低优先级任务的队列。

主题

在中创建队列 AWS PCS

本主题概述了可用选项，并介绍了在中创建队列时应考虑的事项 AWS PCS。

先决条件

- 集 AWS PCS群-只能在与特定集 AWS PCS群关联的情况下创建队列。
- 一个或多个 AWS PCS计算节点组-一个队列必须与至少一个 AWS PCS计算节点组相关联。

要在中创建队列 AWS PCS

您可以使用 AWS Management Console 或创建队列 AWS CLI。

AWS Management Console

使用控制台创建队列

1. 打开控制AWS PCS台。
2. 为队列选择集群。导航到 “队列”，然后选择 “创建队列”。
3. 在队列配置部分中，提供以下值：

    a. 队列名称-队列的名称。名称只能包含字母数字字符（区分大小写）和连字符。它必须以字母字符开头，长度不能超过 25 个字符。该名称在集群中必须是唯一的。

b. 计算节点组-选择 1 个或多个计算节点组来为该队列提供服务。一个计算节点组可以与多个队列关联。

4. （可选）在 “标签” 下，将所有标签添加到 AWS PCS队列中
5. 选择创建队列。创建队列时，“状态” 字段将显示 “正在 AWS PCS创建”。创建队列可能需要几分钟。

建议采取下一步行动

- 向您的新队列提交任务。

AWS CLI

使用创建队列 AWS CLI

使用以下命令创建队列。进行以下替换：

1. Replace（替换）*region-code* 使用集群的 AWS 区域。例如，us-east-1。
2. Replace（替换）*my-queue* 用你的队列的名字。名称只能包含字母数字字符（区分大小写）和连字符。它必须以字母字符开头，长度不能超过 25 个字符。该名称在集群中必须是唯一的。
3. Replace（替换）*my-cluster* 使用您的集群的名称或 ID。
4. Replace（替换）*compute-node-group-id* 带有为队列提供服务的计算节点组的 ID。例如，pcs_abcdef12345。

> **Note**
>
> 创建队列时，必须提供计算节点组的 ID，而不是其名称。

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

创建队列可能需要几分钟。您可以使用以下命令查询队列的状态。在队列状态达到之前，您将无法向队列提交作业ACTIVE。

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

建议采取下一步行动

- 向新队列提交作业

更新队 AWS PCS列

本主题概述了可用选项，并介绍了更新 AWS PCS队列时应考虑的事项。

更新 AWS PCS队列时的注意事项

队列更新不会影响正在运行的作业，但是在队列更新期间，集群可能无法接受新作业。

更新队 AWS PCS列

您可以使用 AWS Management Console 或 AWS CLI 来更新队列。

AWS Management Console

更新队列

1. 从 `https://console.aws.amazon.com/pcs/home#/clusters` 打开 AWS PCS 控制台
2. 选择要在其中更新队列的集群。
3. 导航至 “队列”，转至要更新的队列，然后选择 “编辑”。
4. 在队列配置部分中，更新以下任意值：
   - 节点组-添加或移除计算节点组与队列的关联。
   - 标签-为队列添加或移除标签。
5. 选择更新。应用更改时，“状态” 字段将显示 “正在更新”。

> **Important**
> 队列更新可能需要几分钟。

AWS CLI

更新队列

1. 使用以下命令更新您的队列。在运行命令之前，进行以下替换：

   a. Replace（替换）*region-code* 使用您 AWS 区域 要在其中创建集群的。

   b. Replace（替换）*my-queue* 用你的队列computeNodeGroupId的名字或名字。

   c. Replace（替换）*my-cluster* 使用您的集群clusterId的名称或。

   d. 要更改计算节点组关联，请提供更新后的列表--compute-node-group-configurations。

      - 例如，要添加第二个计算节点组，请执行computeNodeGroupExampleID2以下操作：

        ```
        --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
   ```

2. 更新队列可能需要几分钟。您可以使用以下命令查询队列的状态。在队列状态达到之前，您将无法向队列提交作业ACTIVE。

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

建议的后续步骤

- 向更新后的队列提交任务。

删除中的队列 AWS PCS

本主题概述了如何删除中的队列 AWS PCS。

删除队列时的注意事项

- 如果队列中有作业在运行，则在删除队列时，调度程序将终止这些作业。队列中的待处理任务将被取消。考虑等待队列中的任务完成，或者使用调度程序的本机命令（例如 Slurm）手动停止/取消任务。`scancel`

删除队列

您可以使用 AWS Management Console 或 AWS CLI 删除队列。

AWS Management Console

删除队列

1. 打开控制AWS PCS台。
2. 选择队列的集群。
3. 导航到队列并选择要删除的队列。
4. 选择删除。
5. 将显示 “状态” 字段`Deleting`。可能需要几分钟的时间才能完成。

> **Note**
> 您可以使用调度程序原生的命令来确认队列已删除。例如，对于 Slur `squeue m` 使用`sinfo`或。

AWS CLI

删除队列

- 使用以下命令删除队列，并使用以下替换命令：
  - Replace（替换）*`region-code`* AWS 区域 你的集群就在里面。

- Replace（替换）*my-queue* 使用您的队列的名称或 ID。
- Replace（替换）*my-cluster* 使用您的集群的名称或 ID。

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

删除队列可能需要几分钟。

> **Note**
>
> 您可以使用调度程序原生的命令来确认队列已删除。例如，对于 Slur squeue m 使用sinfo或。

AWS PCS登录节点

一个 AWS PCS集群通常需要至少 1 个登录节点来支持交互式访问和任务管理。实现这一目标的一种方法是为登录节点功能配置静态 AWS PCS计算节点组。您也可以将独立EC2实例配置为登录节点。

主题

使用 AWS PCS计算节点组提供登录节点

本主题概述了建议的配置选项，并介绍了在使用AWSPCS计算节点组为集群提供持久的交互式访问时应考虑的事项。

为登录节点创建 AWS PCS计算节点组

从操作上讲，这与创建常规计算节点组没有太大区别。但是，可以做出一些关键的配置选择：

- 为计算节点组中的至少一个EC2实例设置静态扩展配置。
- 选择按需购买选项，以避免回收您的实例。
- 为计算节点组选择一个信息性名称，例如登录。
- 如果您希望登录节点实例可在您的外部访问VPC，请考虑使用公有子网。
- 如果您打算允许SSH访问，则启动模板需要有一个安全组，该组将SSH端口暴露给您选择的 IP 地址。
- IAM实例配置文件应仅具有您希望最终用户拥有的AWS权限。有关详细信息，请参阅AWS 并行计算服务的 IAM 实例配置文件。
- 考虑允许 AWS Systems Manager 会话管理器管理您的登录实例。
- 考虑将实例AWS凭证的访问权限限制为仅限管理用户
- 选择比普通计算节点组更便宜的实例类型，因为登录节点将持续运行。
- 使用与其他计算节点组相同的（或衍生的）AMI，以帮助确保所有实例都安装了相同的软件。有关自定义的更多信息AMIs，请参阅 Amazon 机器映像 (AMIs) 适用于 AWS PCS
- 在登录节点上配置与计算实例相同的网络文件系统（Amazon EFS、Amazon FSx for Lustre 等）的挂载。有关更多信息，请参阅 将网络文件系统与 AWS PCS。

访问您的登录节点

当您的新计算节点组达到ACTIVE状态后，您可以找到它创建的EC2实例并登录到这些实例中。有关更多信息，请参阅 在中查找计算节点组实例 AWS PCS。

更新登录节点的 AWS PCS计算节点组

您可以使用更新登录节点组 UpdateComputeNodeGroup。作为节点组更新过程的一部分，将替换正在运行的实例。请注意，这将中断实例上所有活跃的用户会话或进程。正在运行或排队的 Slurm 作业不会受到影响。有关更多信息，请参阅 更新 AWS PCS计算节点组。

您也可以编辑计算节点组使用的启动模板。必须使用 UpdateComputeNodeGroup 将更新的启动模板应用于计算节点组。在计算节点组中启动的新EC2实例使用更新的启动模板。有关更多信息，请参阅 将 Amazon EC2 启动模板与 AWS PCS。

删除登录节点的 AWS PCS计算节点组

您可以使用中的删除计算节点组机制更新登录节点组 AWS PCS。作为删除节点组的一部分，正在运行的实例将被终止。请注意，这将中断实例上所有活跃的用户会话或进程。正在运行或排队的 Slurm 作业不会受到影响。有关更多信息，请参阅 删除中的计算节点组 AWS PCS。

使用独立实例作为 AWS PCS登录节点

您可以设置独立EC2实例以与 AWS PCS集群的 Slurm 调度器进行交互。这对于创建登录节点、工作站或专用工作流管理主机非常有用，这些主机可以与 AWS PCS群集配合使用，但在 AWS PCS管理之外运行。为此，每个独立实例必须：

1. 安装兼容的 Slurm 软件版本。
2. 能够连接到 AWS PCS集群的 Slurmctld 终端节点。
3. 使用集群的终端节点和密钥正确配置 Slurm Auth 和 Cred Kiosk Daemon (`sackd`)。 AWS PCS有关更多信息，请参阅 Slurm 文档中的 sackd。

本教程可帮助您配置连接到 AWS PCS集群的独立实例。

目录

步骤 1-检索目标 AWS PCS集群的地址和密钥

使用以下命令检索有关目标 AWS PCS集群 AWS CLI 的详细信息。在运行命令之前，进行以下替换：

- *region-code*替换为目标 AWS 区域 集群的运行位置。
- *cluster-ident*替换为目标集群的名称或标识符

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

该命令将返回类似于此示例的输出。

```
{
    "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": "24.05"
        },
        "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"
            }
        ]
    }
}
```

在此示例中，集群 Slurm 控制器端点的 IP 地址为，`10.3.149.220`并且正在端口上运行。`6817``secretArn`将在后面的步骤中使用来检索集群密钥。IP 地址和端口将在后续步骤中用于配置`sackd`服务。

步骤 2-启动实EC2例

启动 EC2 实例

1. 打开 Amazon EC2 控制台。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在名称和标签部分中，提供实例的名称，例如`PCS-LoginNode`。名称作为资源标签（`Name=PCS-LoginNode`）分配给实例。
4. 在 “应用程序和操作系统映像” 部分，AMI为支持的操作系统选择一个 AWS PCS。有关更多信息，请参阅 支持的操作系统。
5. 在实例类型部分，选择支持的实例类型。有关更多信息，请参阅 支持的实例类型。
6. 在密钥对部分，选择要用于实例的SSH密钥对。
7. 在 “网络设置” 部分：
    - 选择编辑。
        i. 选择您的VPC集 AWS PCS群的。
        ii. 对于防火墙（安全组），请选择选择现有安全组。
            A. 选择一个允许在实例和目标 AWS PCS集群的 Slurm 控制器之间进行流量的安全组。有关更多信息，请参阅 安全组要求和注意事项。

B. （可选）选择允许对您的实例进行入站SSH访问的安全组。

8. 在“存储”部分，根据需要配置存储卷。确保配置足够的空间来安装应用程序和库，以启用您的用例。
9. 在“高级”下，选择允许访问集群密钥的IAM角色。有关更多信息，请参阅 获取 Slurm 集群的秘密。
10. 在“摘要”窗格中，选择“启动实例”。

步骤 3-在实例上安装 Slurm

当实例启动并变为活动状态时，请使用您的首选机制连接到该实例。使用提供的 Slurm 安装程序将 Slurm 安装 AWS 到实例上。有关更多信息，请参阅 Slurm 安装程序。

下载 Slurm 安装程序，将其解压缩，然后使用`installer.sh`脚本安装 Slurm。有关更多信息，请参阅 第 3 步 — 安装 Slurm。

步骤 4-检索和存储集群密钥

这些说明需要 AWS CLI. 有关更多信息，请参阅版本 2 AWS Command Line Interface 用户指南 AWS CLI中的安装或更新到最新版本的。

使用以下命令存储集群密钥。

- 为 Slurm 创建配置目录。

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

- 检索、解码和存储集群密钥。在运行此命令之前，请*region-code*替换为目标集群正在运行的区域，并*secret-arn*替换为在步骤 1 中`secretArn`检索到的值。

```
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**
>
> 在多用户环境中，任何有权访问实例的用户如果能够访问实例元数据服务（IMDS），都可能能够获取集群密钥。反过来，这可能允许他们冒充其他用户。考虑将访问权限限制IMDS为root 用户或管理用户。或者，可以考虑使用另一种不依赖实例配置文件来获取和配置密钥的机制。

- 设置 Slurm 密钥文件的所有权和权限。

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

> **Note**
>
> Slurm 密钥必须归运行sackd服务的用户和群组所有。

步骤 5-配置与 AWS PCS集群的连接

要建立与 AWS PCS集群的连接，请按照以下步骤sackd作为系统服务启动。

1. 使用以下命令为sackd服务设置环境文件。在运行命令之前，请将*ip-address*和*port*替换为步骤 1 中从端点检索到的值。

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

2. 创建用于管理sackd流程的systemd服务文件。

```
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-24.05/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. 设置sackd服务文件的所有权。

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

4. 启用该sackd服务。

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

5. 启动 sackd 服务。

```
sudo systemctl start sackd
```

步骤 6-（可选）测试连接

确认sackd服务正在运行。示例输出如下。如果有错误，它们通常会出现在这里。

```
[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-24.05/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
```

使用 Slurm 客户端命令（例如sinfo和）确认与集群的连接是否正常运行。squeue以下是来自的输出示例sinfo。

```
[root@ip-10-3-27-112 ~]# /opt/aws/pcs/scheduler/slurm-24.05/bin/sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
all up infinite 4 idle~ compute-[1-4]
```

您还应该能够提交工作。例如，类似于此示例的命令将在集群中的 1 个节点上启动交互式作业。

```
/opt/aws/pcs/scheduler/slurm-24.05/bin/srun --nodes=1 -p all --pty bash -i
```

AWS PCS联网

您的 AWS PCS集群是在 Amazon 中创建VPC的。本章包括以下有关集群调度器和节点网络的主题。

除了选择用于启动实例的子网外，您还必须使用EC2启动模板为 AWS PCS计算节点组配置网络。有关启动模板的更多信息，请参阅将 Amazon EC2 启动模板与 AWS PCS。

主题

AWS PCS VPC 和子网要求和注意事项

创建 AWS PCS 集群时，您需要将 VPC 指定为该 VPC 中的子网。本主题概述了您在集群中使用的 VPC 和子网的 AWS PCS 特定要求和注意事项。如果您没有可用于 PC AWS S 的 VPC，则可以使用 AWS提供的 AWS CloudFormation 模板创建 VPC。有关更多信息 VPCs，请参阅 Amazon VPC 用户指南中的虚拟私有云 (VPC)。

VPC 要求和注意事项

在创建集群时，您指定的 VPC 必须满足以下要求和注意事项：

- VPC 必须有足够数量的 IP 地址可用于您要创建的集群、任何节点和其他集群资源。有关更多信息，请参阅 Amazon VPC 用户指南中的您的 VPCs 和子网的 IP 地址。
- VPC 必须具有 DNS 主机名和 DNS 解析支持。否则，节点将无法注册客户集群。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。
- VPC 可能需要使用的 VPC 终端节点 AWS PrivateLink 才能联系 AWS PCS API。有关更多信息，请参阅 Amazon VPC 用户指南 AWS PrivateLink中的使用将您的 VPC 连接到服务。

> **Important**
>
> AWS PCS 不支持具有专用实例租期的 VPC。您用于 AWS PCS 的 VPC 必须使用`default`实例租期。您可以更改现有 VPC 的实例租期。有关更多信息，请参阅 Amazon 弹性计算云用户指南中的更改 VPC 的实例租期。

子网要求和注意事项

创建 Slurm 集群时， AWS PCS 会在您指定的子网中创建一个弹性网络接口 (ENI)。此网络接口支持调度器控制器和客户 VPC 之间的通信。网络接口还使 Slurm 能够与部署在客户账户中的组件进行通信。您只能在创建集群时为集群指定子网。

集群的子网要求

您在创建集群时指定的子网必须满足以下要求：

- 子网必须至少有 1 个 IP 地址才能供 AWS PCS 使用。
- 子网不能位于 AWS Outposts AWS Wavelength、或 AWS 本地区域中。
- 子网可以是公有子网或私有子网。如果可能，我们建议您指定私有子网。公有子网是带有路由表的子网，其中包含通往互联网网关的路由；私有子网是带有路由表的子网，其中不包括通往互联网网关的路由。

节点的子网要求

您可以将节点和其他集群资源部署到创建 AWS PCS 集群时指定的子网，也可以部署到同一 VPC 中的其他子网。

将节点和群集资源部署到的任何子网都必须满足以下要求：

- 您必须确保子网有足够的可用的 IP 地址来部署所有节点和群集资源。
- 如果您计划将节点部署到公有子网，则该子网必须自动分配 IPv4 公有地址。
- 如果您部署节点的子网是私有子网，并且其路由表不包括通往网络地址转换 (NAT) 设备的路由 (IPv4)，请使用 AWS PrivateLink 向客户 VPC 添加 VPC 终端节点。节点联系的所有 AWS 服务都需要 VPC 终端节点。 AWS PCS 唯一需要的端点是允许节点调用`RegisterComputeNodeGroupInstance` API 操作。有关更多信息，请参阅 AWS PCS API 参考RegisterComputeNodeGroupInstance中的。
- 公有或私有子网状态不会影响 AWS PCS；所需的端点必须可以访问。

VPC为您的 AWS PCS集群创建

您可以在 AWS 并行计算服务 (VPC) 中为您的集群创建亚马逊虚拟私有云 (Amazon AWS PCS)。

使用 Amazon 将VPC资源启动VPC到您定义的虚拟网络中。此虚拟网络与您在自己的数据中心中运行的传统网络极为相似。但是，它带有使用 Amazon Web Services 的可扩展基础设施的优势。我们建议您在部署生产VPC集群之前全面了解 Amazon VPC 服务。有关更多信息，请参阅什么是亚马逊VPC？在作者视觉模式下。《亚马逊VPC用户指南》。

集PCS群、节点和支持资源（例如文件系统和目录服务）部署在您的 Amazon 中VPC。如果您想将现有 Amazon VPC 与一起使用PCS，则它必须满足中所述的要求AWS PCS VPC 和子网要求和注意事项。本主题介绍如何使用 AWS提供的 AWS CloudFormation 模板创建VPC符合PCS要求的。部署模板后，您可以查看该模板创建的资源，以确切了解它创建了哪些资源以及这些资源的配置。

先决条件

要VPC为其创建 AmazonPCS，您必须拥有创建亚马逊VPC资源的必要IAM权限。这些资源包括子网VPCs、安全组、路由表和路由，以及互联网和NAT网关。有关更多信息，请参阅 Amazon VPC 用户指南中的使用公VPC有子网创建。要查看亚马逊的完整列表EC2，请参阅《服务授权参考》EC2中的亚马逊操作、资源和条件密钥。

创建 Amazon VPC

VPC通过复制并粘贴URL适合您要使用的 AWS 区域 PCS位置来创建。您也可以下载 AWS CloudFormation 模板并自己将其上传到AWS CloudFormation 控制台。

- 美国东部（弗吉尼亚北部）(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
```

- 美国东部（俄亥俄）(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
```

- 美国西部（俄勒冈）(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
```

- 仅限模板

```
https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/
assets/main.yaml
```

为之创建 Amaz VPC on PCS

1. 在AWS CloudFormation 控制台中打开模板。

   > **Note**
   >
   > 它们已在模板中预先填充，因此您只需将其保留为默认值即可。

2. 在 “提供堆栈名称” 下，然后输入 “堆栈名称” `hpc-networking`。
3. 在参数下，输入以下详细信息：

   a. 然后 VPC，在下CidrBlock方输入 `10.3.0.0/16`

   b. 在子网 A 下：

      i. 然后输入 CidrPublicSubnetA `10.3.0.0/20`

      ii. 然后输入 CidrPrivateSubnetA `10.3.128.0/20`

   c. 在子网 B 下：

      i. 然后输入 CidrPublicSubnetB `10.3.16.0/20`

      ii. 然后输入 CidrPrivateSubnetA `10.3.144.0/20`

   d. 在子网 C 下：

      i. 对于 ProvisionSubnetsC，选择`True`。

> **Note**
>
> 如果您VPC在可用区少于三个的区域中创建，则如果设置为，则此选项将被忽略`True`。

ii. 然后输入 CidrPublicSubnetB `10.3.32.0/20`

iii. 然后输入 CidrPrivateSubnetA `10.3.160.0/20`

4. 在 “权能” 下，选中 “我确认这AWS CloudFormation 可能会创建IAM资源” 复选框。

监控 AWS CloudFormation 堆栈的状态。当它到达时`CREATE_COMPLETE`，VPC资源已准备就绪，可供您使用。

> **Note**
>
> 要查看 AWS CloudFormation 模板创建的所有资源，请打开AWS CloudFormation 控制台。选择 `hpc-networking` 堆栈，然后选择 Resources（资源）选项卡。

中的安全组 AWS PCS

Amazon 中的安全组EC2充当虚拟防火墙，用于控制实例的入站和出站流量。使用 AWS PCS 计算节点组的启动模板向其实例添加或移除安全组。如果您的启动模板不包含任何网络接口，`SecurityGroupIds`请使用提供安全组列表。如果您的启动模板定义了网络接口，则必须使用`Groups`参数为每个网络接口分配安全组。有关启动模板的更多信息，请参阅将 Amazon EC2 启动模板与 AWS PCS。

> **Note**
>
> 对启动模板中安全组配置的更改仅影响在更新计算节点组后启动的新实例。

安全组要求和注意事项

AWS PCS在创建集群时指定的子网中创建跨账户弹性网络接口 (ENI)。这为在由 AWS管理的账户中运行的HPC调度程序提供了与启动的EC2实例进行通信的路径。 AWS PCS您必须为此提供一个安全组ENI，允许调度程序ENI和您的集群实例EC2之间进行双向通信。

实现这一目标的一种直接方法是创建一个允许的自引用安全组，允许该组所有成员之间的所有端口上TCP的 /IP 流量。您可以将其连接到集群和节点组EC2实例。

许可安全组配置示例

Rule type	协议	端口	来源	目标位置
入站	全部	全部	自身	
出站	全部	全部		0.0.0.0/0
出站	全部	全部		自身

这些规则允许所有流量在 Slurm 控制器和节点之间自由流动，允许所有出站流量到达任何目的地，并启用EFA流量。

限制性安全组配置示例

您还可以限制集群与其计算节点之间的开放端口。对于 Slurm 调度程序，连接到集群的安全组必须允许以下端口：

- 6817-启用`slurmctld`从EC2实例到的入站连接
- 6818 — 启用从`slurmctld`到实例上`slurmd`EC2运行的出站连接

连接到计算节点的安全组必须允许以下端口：

- 6817 — 启用`slurmctld`从EC2实例到的出站连接。
- 6818 — 启用与节点组实例之间的入`slurmd`站`slurmctld`到`slurmd`和出站连接
- 60001—63000 — 要支持的节点组实例之间的入站和出站连接 `srun`
- EFA节点组实例之间的流量。有关更多信息，请参阅 Linux 实例用户指南中的准备EFA已启用安全组
- 您的工作负载所需的任何其他节点间流量

里面有多个网络接口 AWS PCS

有些EC2实例有多个网卡。这使他们能够提供更高的网络性能，包括超过 100 Gbps 的带宽能力和改进的数据包处理。有关带有多个网卡的实例的更多信息，请参阅 Amazon 弹性计算云用户指南中的弹性网络接口。

通过向计算节点组的EC2启动模板添加网络接口，为 AWS PCS计算节点组中的实例配置其他网卡。以下是启用两个网卡的启动模板示例，例如可以在hpc7a.96xlarge实例上找到。请注意以下细节：

- 每个网络接口的子网必须与您在配置将使用启动模板的 AWS PCS计算节点组时选择的子网相同。
- 通过将设置为来建立主网络设备，用于进行例行网络通信（例如SSH和HTTPS流量）0。DeviceIndex其他网络接口DeviceIndex有1。只能有一个主网络接口，所有其他接口都是辅助接口。
- 所有网络接口都必须具有唯一性NetworkCardIndex。建议的做法是按照启动模板中定义的顺序对它们进行编号。
- 每个网络接口的安全组都是使用设置的Groups。在此示例中，将入站SSH安全组 (sg-*SshSecurityGroupId*) 添加到主网络接口，以及启用集群内通信的安全组 () sg-*ClusterSecurityGroupId*。最后，在主接口和辅助接口上都添加了一个允许出站连接到 Internet (sg-*InternetOutboundSecurityGroupId*) 的安全组。

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

中的EC2实例的置放群组 AWS PCS

您可以使用置放群组来影响EC2实例的放置，以适应在实例上运行的工作负载的需求。

置放群组类型

- 集群 — 将实例紧密地打包在可用区中，以优化低延迟通信。
- 分区-跨逻辑分区分布实例，以帮助最大限度地提高弹性。
- Sp read — 严格要求少量实例在不同的硬件上启动，这也有助于提高弹性。

有关更多信息，请参阅亚马逊弹性计算云用户指南中的亚马逊EC2实例置放群组。

我们建议您在配置 AWS PCS计算节点组以使用 Elastic Fabric Adapter (EFA) 时加入集群置放群组。

创建与之配合使用的集群置放群组 EFA

1. 为计算节点组创建集群类型的置放群组。
   - 使用以下 AWS CLI 命令：

     ```
     aws ec2 create-placement-group --strategy cluster --group-name PLACEMENT-GROUP-NAME
     ```

   - 您也可以使用 CloudFormation 模板来创建置放群组。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的使用 CloudFormation模板。从以下内容下载模板URL并将其上传到CloudFormation 控制台。

     ```
     https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-placement-group.yaml
     ```

2. 将置放群组包含在 AWS PCS计算节点组的EC2启动模板中。

将弹性织物适配器 (EFA) 与 AWS PCS

Elastic Fabric Adapter (EFA) 是一种高性能的高级网络互连 AWS ，您可以通过它连接到您的EC2实例，以加速高性能计算 (HPC) 和机器学习应用程序。启用在 AWS PCS集群上运行的应用程序EFA需要配置要使用的 AWS PCS计算节点组实例，EFA如下所示。

> **Note**
>
> 安装EFA在 AWS PCS兼容的 AMI — AWS PCS 计算节点组中AMI使用的必须安装并加载EFA驱动程序。有关如何在安装了EFA软件的情况下构建自定义AMI版本的信息，请参阅定制 Amazon 机器映像 (AMIs) 适用于 AWS PCS。

目录

识别EFA已启用的实例 EC2

要使用EFA，AWS PCS计算组允许使用的所有实例类型都必须支持EFA，并且必须具有相同数量的vCPUs（GPUs如果适用）。有关EFA已启用实例的列表，请参阅《亚马逊弹性计算云用户指南》EC2中的亚马逊弹性结构适配HPC器和机器学习工作负载。您还可以使用 AWS CLI 查看支持的实例类型列表EFA。Replace（替换）*region-code* 以及你使用 AWS 区域 的地方 AWS PCS，例如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**
>
> 确定有多少网络接口可用 — 有些EC2实例有多个网卡。这允许他们有多个EFAs。有关更多信息，请参阅 里面有多个网络接口 AWS PCS。

创建安全组以支持EFA通信

AWS CLI

您可以使用以下 AWS CLI 命令创建支持的安全组EFA。该命令输出一个安全组 ID。进行以下替换：

- *region-code*— 指定使用 AWS 区域 地点 AWS PCS，例如us-east-1。

- *vpc-id*— 指定您用于VPC的 ID AWS PCS。
- *efa-group-name*— 提供您为安全组选择的名称。

```
aws ec2 create-security-group \
    --group-name efa-group-name \
    --description "Security group to enable EFA traffic" \
    --vpc-id vpc-id \
    --region region-code
```

使用以下命令附加入站和出站安全组规则。进行以下替换：

- *efa-secgroup-id*— 提供您刚刚创建EFA的安全组的 ID。

```
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

您可以使用 CloudFormation 模板来创建支持的安全组EFA。从以下链接下载模板URL，然后将其上传到AWS CloudFormation 控制台。

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

在 AWS CloudFormation 控制台中打开模板后，输入以下选项。

- 在 “提供堆栈名称” 下
  - 在堆栈名称下，输入一个名称，例如`efa-sg-stack`。
- 在 “参数” 下
  - 在下方 SecurityGroupName，输入一个名称，例如`efa-sg`。
  - 在下方 VPC，选择要使用VPC的地方 AWS PCS。

完成 CloudFormation 堆栈的创建并监控其状态。当它到达CREATE_COMPLETE时，EFA安全组就可以使用了。

（可选）创建置放群组

我们建议您启动集群置放群组EFA中使用的所有实例，以最大限度地缩短它们之间的物理距离。为您计划使用的每个计算节点组创建一个置放群组EFA。中的EC2实例的置放群组 AWS PCS要为您的计算节点组创建置放群组，请参阅。

创建或更新EC2启动模板

EFA网络接口是在 AWS PCS计算节点组的EC2启动模板中设置的。如果有多个网卡，则EFAs可以配置多个网卡。EFA安全组和可选置放群组也包含在启动模板中。

以下是带有两张网卡的实例的启动模板示例，例如 hpc 7a.96xlarge。实例将在集群置放群组subnet-*SubnetID1*中启动pg-*PlacementGroupId1*。

必须专门向每个EFA接口添加安全组。每个人都EFA需要支持EFA流量的安全组(sg-*EfaSecGroupId*)。其他安全组，尤其是处理常规流量（如SSH或）的安全组HTTPS，只需要连接到主网络接口（由 a DeviceIndex 指定0）即可。定义网络接口的启动模板不支持使用SecurityGroupIds参数设置安全组，您必须在配置的每个网络接口Groups中为设置一个值。

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

为以下对象创建或更新计算节点组 EFA

您的 AWS PCS计算节点组必须包含具有相同数量vCPUs、处理器架构和EFA支持的实例。配置计算节点组，使其在AMI安装了EFA软件的情况下使用，并使用配置EFA启用了网络接口的启动模板。

（可选）测试 EFA

通过运行EFA软件安装中包含的fi_pingpong程序，您可以演示计算节点组中两个节点之间已EFA启用通信。如果此测试成功，则很可能配置EFA正确。

要启动，您需要在计算节点组中运行两个实例。如果您的计算节点组使用静态容量，则应该已经有可用的实例。对于使用动态容量的计算节点组，您可以使用salloc命令启动两个节点。以下是一个集群的示例，该群集的动态节点组名为hpc7g与名为的队列相关联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
```

使用找出两个已分配节点的 IP 地址scontrol。在以下示例中，地址分别是 for hpc7g-1 和 10.3.140.69 f 10.3.132.211 or 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=23.11.8
   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=23.11.8
   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
```

使用（或hpc7g-1）连接到其中一个节点SSH（在本例中为SSM）。请注意，这是一个内部 IP 地址，因此，如果您使用，则可能需要从其中一个登录节点进行连接SSH。另请注意，需要通过计算节点组启动模板为实例配置SSH密钥。

```
% ssh ec2-user@10.3.140.69
```

现在，fi_pingpong以服务器模式启动。

```
/opt/amazon/efa/bin/fi_pingpong -p efa
```

Connect 连接到第二个实例 (hpc7g-2)。

```
% ssh ec2-user@10.3.132.211
```

fi_pingpong在客户端模式下运行，连接到服务器hpc7g-1。您应该看到类似于以下示例的输出。

```
% /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
```

（可选）使用 CloudFormation模板创建EFA启用启动模板

由于需要设置多个依赖项EFA，因此提供了一个可用于配置计算节点组的 CloudFormation 模板。它支持最多带有四个网卡的实例。要详细了解带有多个网卡的实例，请参阅 Amazon 弹性计算云用户指南中的弹性网络接口。

从以下位置下载 CloudFormation 模板URL，然后将其上传 AWS 区域 到您使用的 CloudFormation 控制台 AWS PCS。

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/pcs-lt-efa.yaml
```

在 AWS CloudFormation 控制台中打开模板后，输入以下值。请注意，模板将提供一些默认参数值，您可以将其保留为默认值。

- 在 “提供堆栈名称” 下
  - 在堆栈名称下，输入描述性名称。我们建议使用您将为 AWS PCS计算节点组选择的名称，例如*NODEGROUPNAME*-efa-lt。
- 在 “参数” 下
  - 在下方 NumberOfNetworkCards，选择您的节点组中实例中的网卡数量。
  - 在下方 VpcId，选择VPC AWS PCS集群的部署位置。
  - 在下方 NodeGroupSubnetId，选择集群中将在VPC其中启动EFA启用了的实例的子网。
  - 在下方 PlacementGroupName，将该字段留空，为该节点组创建新的集群置放群组。如果您要使用现有的置放群组，请在此处输入其名称。

  - 在下 ClusterSecurityGroupId，选择您要使用的安全组，以允许访问集群中的其他实例和 AWS PCSAPI。许多客户从其集群中选择默认安全组VPC。
  - 在下方 SshSecurityGroupId，提供您用于允许对集群中节点进行入站SSH访问的安全组的 ID。
  - 对于 SshKeyName，选择用于访问集群中节点的SSH密钥对。
  - 对于 LaunchTemplateName，输入启动模板的描述性名称，例如*NODEGROUPNAME*`-efa-lt`。在您要使用的 AWS 区域 位置 AWS 账户 中，该名称必须是唯一的 AWS PCS。
- 能力不足
  - 选中 “我确认这 AWS CloudFormation 可能会创建IAM资源” 复选框。

监控 CloudFormation 堆栈的状态。当它到达`CREATE_COMPLETE`时，启动模板就可以使用了。将其与AWS PCS计算节点组一起使用，如上所述为以下对象创建或更新计算节点组 EFA。

将网络文件系统与 AWS PCS

您可以将网络文件系统附加到 AWS 并行计算服务 (AWS PCS) 计算节点组中启动的节点，以提供写入和访问数据和文件的永久位置。您可以使用 AWS 服务提供的文件系统，包括亚马逊 Elastic File System（亚马逊EFS）、Amazon for O FSx pen ZFS、Amazon FSx for Lustre 和亚马逊文件缓存。您也可以使用自行管理的文件系统，例如NFS服务器。

本主题介绍使用网络文件系统的注意事项和示例 AWS PCS。

使用网络文件系统的注意事项

各种文件系统的实现细节各不相同，但有一些常见的注意事项。

- 必须在实例上安装相关的文件系统软件。例如，要使用 Amazon f FSx or Lustre，相应的 Lustre 包裹应该在场。这可以通过将其包含在计算节点组中AMI或使用在实例启动时运行的脚本来实现。
- 共享网络文件系统和计算节点组实例之间必须有网络路由。
- 共享网络文件系统和计算节点组实例的安全组规则必须允许连接到相关端口。
- 你必须保持一致 POSIX 跨访问文件系统的资源中的用户和组命名空间。否则，在您的PCS集群上运行的作业和交互式进程可能会遇到权限错误。
- 文件系统装载是使用完成的 EC2 启动模板。挂载网络文件系统时出现错误或超时可能会使实例无法运行作业。反过来，这可能会导致意想不到的成本。有关调试启动模板的更多信息，请参阅将 Amazon EC2 启动模板与 AWS PCS。

网络挂载示例

您可以使用 Amazon EFS、Amazon for Lustre、Amazon FSx for Open ZFS 和 Amazon FSx File Cache 创建文件系统。展开下面的相关部分，查看每个网络挂载的示例。

Amazon EFS

文件系统设置

创建 Amazon EFS 文件系统。确保它在每个可用区中都有一个挂载目标，您将在其中启动PCS计算节点组实例。还要确保每个挂载目标都与一个安全组相关联，该安全组允许来自PCS计算节点组实例的入站和出站访问。有关更多信息，请参阅 Amazon Elastic File System 用户指南中的挂载目标和安全组。

启动模板

将文件系统设置中的安全组添加到将用于计算节点组的启动模板中。

包括使用挂载 Amazon EFS 文件系统的`cloud-config`机制的用户数据。用您自己的详细信息替换此脚本中的以下值：

- *`mount-point-directory`*— 您将在每个实例上挂载 Amazon 的路径 EFS
- *`filesystem-id`*— 文件系统的EFS文件系统 ID

```
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==--
```

亚马逊 f FSx or Lustre

文件系统设置

在你要使用的VPC AWS PCS位置创建一个 FSx for Lustre 文件系统。为了最大限度地减少区域间传输，请在同一个可用区的子网中部署，您将在那里启动大多数PCS计算节点组实例。确保文件系统与允许来自PCS计算节点组实例的入站和出站访问的安全组相关联。有关安全组的更多信息，请参阅《Amazon for Lustre 用户指南》VPC中的 Amazon FSx 文件系统访问控制。

启动模板

包括FSx用于装载 for `cloud-config` Lustre 文件系统的用户数据。用您自己的详细信息替换此脚本中的以下值：

- *`mount-point-directory`*— 你要为 Lustre 挂载FSx的实例上的路径

- *filesystem-id*— 适用于 Lustre 文件系统的文件系统 ID FSx
- *mount-name*— 适用于 Lustre 文件FSx系统的装载名称
- *region-code*— for FSx Lustre 文件系统的部署 AWS 区域 位置（必须与您的 AWS PCS系统相同）
- （可选）*latest*-任何版本的 Lustre 由 for Lu FSx stre 支持

```
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==
```

Amaz FSx on 公开版 ZFS

文件系统设置

在你要使用的VPC位置创建一个 FSx for Open ZFS 文件系统 AWS PCS。为了最大限度地减少区域间传输，请在同一个可用区的子网中部署，您将在那里启动大多数 AWS PCS计算节点组实例。确保文件系统与允许从 AWS PCS计算节点组实例进行入站和出站访问的安全组相关联。有关安全组的更多信息，请参阅《开放ZFS用户指南》VPC中的 “FSx使用 Amazon 管理文件系统访问权限”。

启动模板

包括用于挂载 fo `cloud-config` r Open ZFS 文件系统的根卷FSx的用户数据。用您自己的详细信息替换此脚本中的以下值：

- *mount-point-directory*— 您要在实例上挂载 for Open ZFS 共享FSx的路径
- *filesystem-id*— Open ZFS 文件系统的文件系统 ID FSx
- *region-code*— f FSx or Open ZFS 文件系统的部署 AWS 区域 位置（必须与您的 AWS PCS系统相同）

```
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

文件系统设置

在您要使用的VPC位置创建 Amazon 文件缓存 AWS PCS。要最大限度地减少区域间传输，请在要启动大部分PCS计算节点组实例的同一可用区中选择一个子网。确保文件缓存与安全组关联，该安全组允许您的PCS实例和文件缓存之间通过端口 988 进行入站和出站流量。有关安全组的更多信息，请参阅《Amazon 文件缓存用户指南》VPC中的 Amazon 缓存访问控制。

启动模板

将文件系统设置中的安全组添加到将用于计算节点组的启动模板中。

包括用于cloud-config挂载 Amazon 文件缓存的用户数据。用您自己的详细信息替换此脚本中的以下值：

- *mount-point-directory*— 你要为 Lustre 挂载FSx的实例上的路径
- *cache-dns-name*— 文件缓存的域名系统 (DNS) 名称
- *mount-name*— 文件缓存的挂载名称

```
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 机器映像 (AMIs) 适用于 AWS PCS

AWS PCS与您AMIs提供的软件配合使用，为集群中节点上的软件和配置提供了极大的灵活性。如果您正在尝试 AWS PCS，则可以使用由AMI提供并由维护的示例 AWS。如果您 AWS PCS在生产环境中使用，我们建议您自己构建AMIs。本主题介绍如何发现和使用示例AMIs，以及如何构建和使用自己的自定义示例AMIs。

主题

使用示例 Amazon 系统映像 (AMIs) AWS PCS

AWS提供了AMIs可以用作工作起点的示例 AWS PCS。

> **Important**
>
> 示例AMIs仅用于演示目的，不建议用于生产工作负载。

查找当前 AWS PCS样本 AMIs

AWS Management Console

AWSPCS示例AMIs具有以下命名约定：

```
aws-pcs-sample_ami-OS-architecture-scheduler-scheduler-major-version
```

接受的值

- *OS* – amzn2
- *architecture* – x86_64 或 arm64
- *scheduler* – slurm
- *scheduler-major-version* – 24.05

要查找 AWS PCS样品 AMIs

1. 打开 Amazon EC2 控制台。
2. 导航到 AMIs。
3. 选择公有映像。
4. 在AMI按属性或标签查找中，AMI使用模板化名称搜索。

   示例

   - Arm64 实例上AMI的 Slurm 24.05 示例

     ```
     aws-pcs-sample_ami-amzn2-arm64-slurm-24.05
     ```

   - x86 实例AMI上的 Slurm 24.05 示例

     ```
     aws-pcs-sample_ami-amzn2-x86_64-slurm-24.05
     ```

   > **Note**
   >
   > 如果有多个AMIs，请使用AMI带有最新时间戳的。

5. 创建或更新计算节点组时使用 AMI ID。

AWS CLI

您可以通过以下命令找到最新的 AWS PCS示例AMI。*region-code*替换为你使用 AWS 区域 的地方 AWS PCS，例如`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-24.05*' \
            'Name=state,Values=available' \
  --query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
  ```

- Arm64

  ```
  aws ec2 describe-images --region region-code --owners amazon \
  --filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-arm64-slurm-24.05*' \
  ```

```
            'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

创建或更新计算节点组时使用 AMI ID。

了解有关 AWS PCS样品的更多信息 AMIs

要查看该 AWS PCS示例当前版本和先前版本的内容和配置详细信息AMIs，请参见AWS PCS示例的发行说明 AMIs。

自己动手搭建AMIs兼容 AWS PCS

要了解如何自己构建可与之配合使用 AMIs AWS PCS，请参阅定制 Amazon 机器映像 (AMIs) 适用于 AWS PCS。

定制 Amazon 机器映像 (AMIs) 适用于 AWS PCS

AWS PCS旨在与您带到服务中的 Amazon 系统映像 (AMI) 配合使用。它们上面AMIs可以安装任意软件和配置，只要它们安装并正确配置了 AWS PCS代理和兼容版本的 Slurm 即可。您必须使用 AWS提供的安装程序在您的自定义安装程序上安装该 AWS PCS软件。AMI我们建议你使用 AWS提供的安装程序在你的自定义安装程序上安装 Slurm，AMI但如果你愿意，你可以自己安装 Slurm（不推荐）。

> **Note**
>
> 如果您想在 AWS PCS不构建自定义项的情况下进行尝试AMI，则可以使用AMI提供的示例 AWS。有关更多信息，请参阅 使用示例 Amazon 系统映像 (AMIs) AWS PCS。

本教程将帮助您创建可与PCS计算节点组配合使用的AMI，为您的工作负载HPC和 AI/ML 工作负载提供支持。

主题

- 步骤 1-启动临时实例
- 步骤 2-安装代 AWS PCS理
- 第 3 步 — 安装 Slurm
- 步骤 4-（可选）安装其他驱动程序、库和应用程序软件
- 第 5 步 — 创建与之AMI兼容的 AWS PCS

步骤 1-启动临时实例

启动一个临时实例，您可以使用该实例来安装和配置 AWS PCS软件和 Slurm 调度程序。您可以使用此实例来创建与AMI兼容的 AWS PCS。

启动临时实例

1. 打开亚马逊EC2控制台。
2. 在导航窗格中，选择实例，然后选择启动实例以打开新的启动实例向导。
3. （可选）在名称和标签部分中，提供实例的名称，例如`PCS-AMI-instance`。名称作为资源标签（`Name=PCS-AMI-instance`）分配给实例。
4. 在 “应用程序和操作系统映像” 部分，AMI为支持的操作系统选择一个。
5. 在 Instance type（实例类型）部分中，选择 supported instance type（支持的实例类型）。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 “网络设置” 部分：
    - 对于防火墙（安全组），选择选择现有安全组，然后选择允许入站SSH访问您的实例的安全组。
8. 在 Storage（存储）部分中，根据需要配置卷。确保配置足够的空间来安装您自己的应用程序和库。
9. 在 Summary（摘要）面板中，选择 Launch instance（启动实例）。

步骤 2-安装代 AWS PCS理

安装用于配置由 Slurm 启动的实例 AWS PCS的代理。

安装 AWS PCS 代理

1. 连接到您启动的实例。有关更多信息，请参阅连接到您的 Linux 实例。
2. （可选）为确保您的所有软件包都是最新的，请对您的实例执行快速软件更新。此过程可能需要几分钟时间。
    - 亚马逊 Linux 2、RHEL 9、Rocky Linux 9

```
sudo yum update -y
```

- Ubuntu 22.04

```
sudo apt-get update && sudo apt-get upgrade -y
```

3. 重启实例并重新连接到它。
4. 下载 AWS PCS代理安装文件。安装文件被打包成压缩的 tarball (`.tar.gz`) 文件。要下载最新的稳定 版本，请使用以下命令。*region*替换为启动临时实例 AWS 区域 的位置，例如`us-east-1`。

```
curl https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz -o aws-pcs-agent-v1.1.1-1.tar.gz
```

您也可以通过将前面的命令`latest`中的版本号替换为来获取最新版本（例如：`aws-pcs-agent-v1-latest.tar.gz`）。

> **Note**
>
> 在 future 版本的 AWS PCS代理软件中，这种情况可能会发生变化。

5. （可选）验证 AWS PCS软件压缩包的真实性和完整性。建议您执行此操作以验证软件发布者的身份，并检查该文件自发布以来是否已被更改或损坏。

   a. 下载的公GPG钥 AWS PCS并将其导入您的密钥环。*region*替换为启动临时实例 AWS 区域的位置。该命令应返回一个密钥值。记录密钥值；您可以在下一步中使用它。

   ```
   wget https://aws-pcs-repo-public-keys-region.s3.amazonaws.com/aws-pcs-public-key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. 运行以下命令以验证GPG密钥的指纹。

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   该命令应返回与以下内容相同的指纹：

   ```
   1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   ```

> **Important**
>
> 如果指纹不匹配，请不要运行 AWS PCS代理安装脚本。请联系 AWS Support。

c. 下载签名文件并验证 AWS PCS软件 tarball 文件的签名。*region*替换为您启动临时实例 AWS 区域 的位置，例如us-east-1。

```
wget https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-agent-v1.1.1-1.tar.gz.sig
```

该输出应该类似于以下内容：

```
gpg: assuming signed data in './aws-pcs-agent-v1.1.1-1.tar.gz'
gpg: Signature made Fri Dec 13 18:50:19 2024 CEST
gpg:                using RSA key 4BAA531875430EB0739E6D961BA7F0AF6E34C496
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)" [unknown]
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: 4BAA 5318 7543 0EB0 739E  6D96 1BA7 F0AF 6E34 C496
```

如果结果包含Good signature并且指纹与上一步返回的指纹相匹配，则继续下一步。

> **Important**
>
> 如果指纹不匹配，请不要运行 AWS PCS软件安装脚本。请联系 AWS Support。

6. 从压缩文件中提取.tar.gz文件并导航到解压缩的目录。

```
tar -xf aws-pcs-agent-v1.1.1-1.tar.gz && \
    cd aws-pcs-agent
```

7. 安装 AWS PCS 软件。

```
sudo ./installer.sh
```

8. 检查 AWS PCS软件版本文件以确认安装成功。

```
cat /opt/aws/pcs/version
```

该输出应该类似于以下内容：

```
AGENT_INSTALL_DATE='Fri Dec 13 12:28:43 UTC 2024'
AGENT_VERSION='1.1.1'
AGENT_RELEASE='1'
```

第 3 步 — 安装 Slurm

安装与兼容的 Slurm 版本。 AWS PCS

> **Note**
>
> 如果您安装了AMI先前版本的 Slurm 软件，则必须执行以下步骤才能安装新版本的 Slurm。根据创建集群时配置的 Slurm 版本， AWS PCS代理会在运行时启用 Slurm 二进制文件的正确版本。

要安装 Slurm

1. Connect 连接到安装 AWS PCS软件的同一个临时实例。
2. 下载 Slurm 安装程序软件。Slurm 安装程序被打包成压缩的 tarball () `.tar.gz` 文件。要下载最新的稳定 版本，请使用以下命令。*region*替换 AWS 区域 为临时实例的，例如`us-east-1`。

   ```
   curl https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-slurm/aws-pcs-
   slurm-24.05-installer-24.05.5-2.tar.gz \
       -o aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz
   ```

   您也可以通过将前面的命令`latest`中的版本号替换为来获取最新版本（例如：`aws-pcs-slurm-24.05-installer-latest.tar.gz`）。

   > **Note**
   >
   > 在 Slurm 安装程序软件的未来版本中，这种情况可能会发生变化。

3. （可选）验证 Slurm 安装程序压缩包的真实性和完整性。建议您执行此操作以验证软件发布者的身份，并检查该文件自发布以来是否已被更改或损坏。

   a. 下载的公GPG钥 AWS PCS并将其导入您的密钥环。*region*替换为启动临时实例 AWS 区域的位置。该命令应返回一个密钥值。记录密钥值；您可以在下一步中使用它。

   ```
   wget https://aws-pcs-repo-public-keys-region.s3.amazonaws.com/aws-pcs-public-key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. 运行以下命令以验证GPG密钥的指纹。

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   该命令应返回与以下内容相同的指纹：

   ```
   1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   ```

   > **Important**
   >
   > 如果指纹不匹配，请不要运行 Slurm 安装脚本。请联系 AWS Support。

   c. 下载签名文件并验证 Slurm 安装程序压缩包文件的签名。*region*替换为您启动临时实例 AWS 区域 的位置，例如`us-east-1`。

   ```
   wget https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz.sig && \
       gpg --verify ./aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz.sig
   ```

   该输出应该类似于以下内容：

   ```
   gpg: assuming signed data in './aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz'
   gpg: Signature made Wed Dec 18 14:23:38 2024 CEST
   gpg:                using RSA key 4BAA531875430EB0739E6D961BA7F0AF6E34C496
   gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)" [unknown]
   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: 4BAA 5318 7543 0EB0 739E  6D96 1BA7 F0AF 6E34 C496
   ```

如果结果包含Good signature并且指纹与上一步返回的指纹相匹配，则继续下一步。

> **Important**
>
> 如果指纹不匹配，请不要运行 Slurm 安装脚本。请联系 AWS Support。

4. 从压缩的 .tar.gz 文件中提取文件，并导航到提取的目录。

```
tar -xf aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz && \
    cd aws-pcs-slurm-24.05-installer
```

5. 安装 Slurm。安装程序下载、编译和安装 Slurm 及其依赖项。这需要几分钟，具体取决于您选择的临时实例的规格。

```
sudo ./installer.sh -y
```

6. 检查调度程序版本文件以确认安装。

```
cat /opt/aws/pcs/scheduler/slurm-24.05/version
```

该输出应该类似于以下内容：

```
SLURM_INSTALL_DATE='Wed Dec 18 12:38:56 UTC 2024'
SLURM_VERSION='24.05.5'
PCS_SLURM_RELEASE='2'
```

步骤 4-（可选）安装其他驱动程序、库和应用程序软件

在临时实例上安装其他驱动程序、库和应用程序软件。安装过程将因特定的应用程序和库而异。如果您AWS PCS以前没有构建过自定义AMI版本，我们建议您首先在仅安装 AWS PCS软件和 Slurm 的情况下构建和测试，然后在确认初步成功后逐步添加自己的软件和配置。AMI

示例

- 弹性结构适配器 (EFA) 软件。有关更多信息，请参阅《亚马逊弹性计算云用户指南》EC2中的亚马逊HPC工作负载入门EFA和MPI操作指南。
- 亚马逊 Elastic File System（亚马逊EFS）客户端。有关更多信息，请参阅亚马逊 Elasti c File System 用户指南中的手动安装亚马逊EFS客户端。

- Lustre 客户端，使用亚马逊获取 FSx Lustre 和亚马逊文件缓存。有关更多信息，请参阅 for Lustre 用户指南中的安装 Lustre 客户端。FSx
- Amazon CloudWatch 代理，用于使用 CloudWatch 日志和指标。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的安装 CloudWatch 代理。
- AWS 神经元，使用 t rn* 和 inf * 实例类型。有关更多信息，请参阅 AWS Neuron 文档。
- NVIDIA驱动程序CUDA、和DCGM，用于使用 p* 或 g* 实例类型。

第 5 步 — 创建与之AMI兼容的 AWS PCS

安装所需的软件组件后，您可以创建一个AMI，以便在 AWS PCS计算节点组中重新启动实例。

AMI从您的临时实例创建

1. 打开亚马逊EC2控制台。
2. 在导航窗格中，选择实例。
3. 选择您创建的临时实例。选择 “操作”、“图像”、“创建图像”。
4. 对于 Create image (创建映像)，请执行以下操作：
   a. 在图像名称中，输入描述性名称。AMI
   b. （可选）在图像描述中，输入对图片用途的简要描述AMI。
   c. 选择创建映像。
5. 在导航窗格中，选择 AMIs。
6. 在AMI列表中找到您创建的。等待其状态从 “待定” 变为 “可用”，然后将其与 AWS PCS计算节点组一起使用。

步骤 6-将自定义AMI与 AWS PCS计算节点组配合使用

您可以将自定义节点组AMI与新的或现有的 AWS PCS计算节点组一起使用。

New compute node group

要使用自定义 AMI

1. 打开 AWS PCS 管理控制台。
2. 在导航窗格中，选择集群。

3. 选择要在其中使用自定义的集群AMI，然后选择计算节点组。
4. 创建新的计算节点组。有关更多信息，请参阅 在中创建计算节点组 AWS PCS。在 AMIID 下，搜索AMI要使用的自定义项的名称或 ID。完成计算节点组的配置，然后选择创建计算节点组。
5. （可选）确认AMI支持的实例已启动。在计算节点组中启动实例。为此，您可以将计算节点组配置为具有单个静态实例，也可以向使用该计算节点组的队列提交作业。

   a. 检查 Amazon EC2 控制台，直到实例显示为带有新计算节点组 ID 的标签。有关这方面的更多信息，请参阅在中查找计算节点组实例 AWS PCS...

   b. 当您看到实例启动并完成其引导过程时，请确认它正在使用预期AMI的过程。为此，请选择实例，然后在 “详细信息” 下检查 AMIID。它应与AMI您在计算节点组设置中配置的相匹配。

   c. （可选）将计算节点组扩展配置更新为您的首选值。

Existing compute node group

要使用自定义 AMI

1. 打开 AWS PCS 管理控制台。
2. 在导航窗格中，选择集群。
3. 选择要在其中使用自定义的集群AMI，然后选择计算节点组。
4. 选择要配置的节点组，然后选择编辑。在 AMIID 下，搜索AMI要使用的自定义项的名称或 ID。完成计算节点组的配置，然后选择更新。在计算节点组中启动的新实例将使用更新后的 AMI ID。现有实例将继续使用旧实例，AMI直到 AWS PCS取而代之。有关更多信息，请参阅 更新 AWS PCS计算节点组。
5. （可选）确认AMI支持的实例已启动。在计算节点组中启动实例。为此，您可以将计算节点组配置为具有单个静态实例，也可以向使用该计算节点组的队列提交作业。

   a. 检查 Amazon EC2 控制台，直到实例显示为带有新计算节点组 ID 的标签。有关这方面的更多信息，请参阅在中查找计算节点组实例 AWS PCS...

   b. 当您看到实例启动并完成其引导过程时，请确认它正在使用预期AMI的过程。为此，请选择实例，然后在 “详细信息” 下检查 AMIID。它应与AMI您在计算节点组设置中配置的相匹配。

   c. （可选）将计算节点组扩展配置更新为您的首选值。

步骤 7-终止临时实例

在您确认按预期AMI运行后 AWS PCS，您可以终止临时实例以停止为此收取费用。

终止临时实例

1. 打开亚马逊EC2控制台。
2. 在导航窗格中，选择实例。
3. 选择您创建的临时实例，然后选择操作、实例状态、终止实例。
4. 当系统提示您确认时，选择终止。

要为其定制AMIs构建的软件安装程序 AWS PCS

AWS 提供了可在实例上安装 AWS PCS软件的可下载文件。 AWS 还提供了可以下载、编译和安装相关版本的 Slurm 及其依赖项的软件。您可以使用这些说明来构建自定义版本以AMIs供使用， AWS PCS也可以使用自己的方法。

目录

AWS PCS软件安装程序

AWS PCS软件安装程序将实例配置为在实例引导 AWS PCS过程中使用。您必须使用 AWS提供的安装程序在您的自定义安装程序上安装该 AWS PCS软件。AMI

Slurm 安装程序

Slurm 安装程序下载、编译和安装 Slurm 及其依赖项的相关版本。你可以使用 Slurm 安装程序为其构建自定义AMIs版本。 AWS PCS您也可以使用自己的机制，前提是它们与 Slurm 安装程序提供的软件配置一致。

AWS提供的软件将安装以下内容：

- Slurm 处于请求的主版本和维护版本（当前版本 24.05.x）-许可证 2 GPL
  - Slurm 的构建设置为 `--sysconfdir /etc/slurm`
  - Slurm 是用以下选项构建的 `--enable-pam --without-munge`
  - Slurm 是用选项构建的 `--sharedstatedir=/run/slurm/`
  - Slurm 是用和PMIX支持构建的 JWT
  - Slurm 安装在 `/opt/aws/pcs/schedulers/slurm-24.05`
- 打开 PMIX（版本 4.2.6）— 许可证
  - Op PMIX en 是作为子目录安装的 `/opt/aws/pcs/scheduler/`
- libjwt（版本 1.17.0）— 许可证 -2.0 MPL
  - libjwt 是作为子目录安装的 `/opt/aws/pcs/scheduler/`

AWS提供的软件按如下方式更改系统配置：

- 将版本创建的 Slurm `systemd` 文件复制到文件`/etc/systemd/system/`名中。`slurmd-24.05.service`
- 如果它们不存在，则使用UID/GID创建一个 Slurm 用户和组 (`slurm:slurm`)。`401`
- 在 Amazon Linux 2 和 Rocky Linux 9 上，安装会添加EPEL存储库，用于安装构建 Slurm 或其依赖项所需的软件。
- 安装RHEL9时将启用`codeready-builder-for-rhel-9-rhui-rpms`并`epel-release-latest-9`从中`fedoraproject`安装构建 Slurm 或其依赖项所需的软件。

支持的操作系统

该 AWS PCS软件和 Slurm 安装程序支持以下操作系统：

- Amazon Linux 2
- RedHat 企业 Linux 9
- Rocky Linux 9
- Ubuntu 22.04

有关更多信息，请参阅 中支持的操作系统 AWS PCS。

> **Note**
>
> AWS Deep Learning AMIs (DLAMI) 基于亚马逊 Linux 2 和 Ubuntu 22.04 的版本应与该 AWS PCS软件和 Slurm 安装程序兼容。有关更多信息，请参阅《AWS Deep Learning AMIs 开发者指南》DLAMI中的 “选择你的”。

支持的实例类型

AWS PCS软件和 Slurm 安装程序支持任何可以运行支持的操作系统之一的 x86_64 或 arm64 实例类型。

支持的 Slurm 版本

支持以下主要版本的 Slurm：

- Slurm 24.05
- Slurm 23.11

使用校验和验证安装程序

您可以使用SHA256校验和来验证安装程序压缩包 (.tar.gz) 文件。建议您执行此操作以验证软件发布者的身份，并检查该应用程序自发布以来是否已被更改或损坏。

验证压缩包

使用 s ha256sum 实用程序获取SHA256校验和并指定压缩包文件名。您必须从保存 tarball 文件的目录中运行该命令。

- SHA256

```
$ sha256sum tarball_filename.tar.gz
```

该命令应返回以下格式的校验和值。

```
checksum_value tarball_filename.tar.gz
```

将命令返回的校验和值与下表中提供的校验和值进行比较。如果校验和匹配，则可以安全地运行安装脚本。

> **Important**
>
> 如果校验和不匹配，请不要运行安装脚本。联系 Support。

例如，以下命令生成 Slurm 24.0 SHA256 5.5-2 压缩包的校验和。

```
$ sha256sum aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz
```

输出示例：

```
7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz
```

下表列出了最新版本安装程序的校验和。*us-east-1*替换为你使用 AWS 区域 的地方 AWS PCS。

AWS PCS 代理

Installer（安装程序）	下载 URL	SHA256校验和
AWS PCS特工 1.1.1-1	`https://aws-pcs-repo-us-east-1.s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz`	`bef078bf60a6d8ecde2e6c49cd34d088703f02550279e3bf483d57a235334dc6`
AWS PCS特工 1.1.0-1	`https://aws-pcs-repo-us-east-1.s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.0-1.tar.gz`	`594c32194c71bccc5d66e5213213ae38dd2c6d2f9a950bb01accea0bbab0873a`
AWS PCS特工 1.0.1-1	`https://aws-pcs-repo-us-east-1.s3.amazonaws.com/aws-pcs-a`	`04e22264019837e3f42d8346daf5886eaacecd21571742eb505ea8911786bcb2`

Installer（安装程序）	下载 URL	SHA256校验和
	gent/aws-pcs-agent-v1.0.1-1.tar.gz	
AWS PCS特工 1.0.0-1	https://aws-pcs-repo- *us-east-1* .s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz	d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042144b134ce0

Slurm 安装程序

Installer（安装程序）	下载 URL	SHA256校验和
Slurm 24.05.5-2	https://aws-pcs-repo- *us-east-1* .s3.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz	7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b
Slurm 23.11.10-3	https://aws-pcs-repo- *us-east-1* .s3.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-3.tar.gz	488a10ee0fbd57ec0e0ff7ea708a9e3038fafdc025c6bb391c75c2e2a7852a00
Slurm 23.11.10-2	https://aws-pcs-repo- *us-east-1* .s3.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-2.tar.gz	0bbe85423305c05987931168caf98da08a34c25f9eec0690e8e74de0b7bc8752
Slurm 23.11.10-1	https://aws-pcs-repo- *us-east-1* .s3.amazo	27e8faa9980e92cdfd8cfdc71f937777f093

Installer（安装程序）	下载 URL	SHA256校验和
	naws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-1.tar.gz	4552ce61e33dac4ecf5a20321e44
Slurm 23.11.9-1	https://aws-pcs-repo- *us-east-1* .s3.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz	1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8

AWS PCS示例的发行说明 AMIs

AMIs对于支持的最新计划程序主要版本，请接收安全更新和严重错误修复。这些增量安全补丁未包含在官方发行说明中。

Important

不支持与旧调度程序版本AMIs相关的示例，也不会收到更新。

Important

示例AMIs仅用于演示目的，不建议用于生产工作负载。

目录

AWS PCSx86_64（亚马逊 Linux 2）的示例 AMIs

Slurm 24.05

AMI 名称

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.05`

支持的EC2实例

- 所有采用 64 位 x86 处理器的实例。要查找兼容的实例，请导航至 Amazon EC2 控制台。选择实例类型，然后搜索`Architectures=x86_64`。

AMI内容

- 支持的 AWS 服务： AWS PCS
- 操作系统：亚马逊 Linux 2
- 计算架构：x86_64
- EBS音量类型:gp2
- EFA安装程序：1.33.0
- GDRCopy: 2.4
- NVIDIA驱动程序：550.127.08
- NVIDIACUDA: 12.4.1_550.54.15

Slurm 23.11

AMI 名称

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`

支持的EC2实例

- 所有采用 64 位 x86 处理器的实例。要查找兼容的实例，请导航至 Amazon EC2 控制台。选择实例类型，然后搜索`Architectures=x86_64`。

AMI内容

- 支持的 AWS 服务： AWS PCS
- 操作系统：亚马逊 Linux 2
- 计算架构：x86_64
- EBS音量类型:gp2
- EFA安装程序：1.33.0
- GDRCopy: 2.4
- NVIDIA驱动程序：550.127.08
- NVIDIACUDA: 12.4.1_550.54.15

AWS PCSArm64（亚马逊 Linux 2）的示例 AMIs

Slurm 24.05

AMI 名称

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.05`

支持的EC2实例

- 所有带有 64 位 Arm 处理器的实例。要查找兼容的实例，请导航至 Amazon EC2 控制台。选择实例类型，然后搜索`Architectures=arm64`。

AMI内容

- 支持的 AWS 服务： AWS PCS
- 操作系统：亚马逊 Linux 2
- 计算架构：arm64
- EBS音量类型:gp2
- EFA安装程序：1.33.0
- GDRCopy: 2.4
- NVIDIA驱动程序：550.127.08
- NVIDIACUDA: 12.4.1_550.54.15

Slurm 23.11

AMI 名称

- `aws-pcs-sample_ami-amzn2-arm64-slurm-23.11`

支持的EC2实例

- 所有带有 64 位 Arm 处理器的实例。要查找兼容的实例，请导航至 Amazon EC2 控制台。选择实例类型，然后搜索`Architectures=arm64`。

AMI内容

- 支持的 AWS 服务： AWS PCS
- 操作系统：亚马逊 Linux 2
- 计算架构：arm64
- EBS音量类型:gp2
- EFA安装程序：1.33.0
- GDRCopy: 2.4
- NVIDIA驱动程序：550.127.08
- NVIDIACUDA: 12.4.1_550.54.15

中支持的操作系统 AWS PCS

AWS PCS使用为计算节点组配置的 Amazon 系统映像 (AMI) 启动该计算节点组中的EC2实例。AMI决定了EC2实例使用的操作系统。您无法在 AWS PCS示例中更改操作系统AMIs。AMI如果要使用其他操作系统，则必须创建自定义操作系统。有关更多信息，请参阅 Amazon 机器映像 (AMIs) 适用于 AWS PCS。

支持的操作系统

- Amazon Linux 2

  这是 AWS PCS示例中的操作系统AMIs。

  > **Important**
  >
  > 示例AMIs仅用于演示目的，不建议用于生产工作负载。即使您打算使用 Amazon Linux 2，您也应该AMI为生产工作负载创建和使用自定义。

- RedHat 企业 Linux 9 (RHEL9)

  RHEL任何实例类型的按需成本都高于其他支持的操作系统。有关定价的更多信息，请参阅按需定价和 Amazon Elastic Compute Cloud 上如何提供和定价 Red Hat Enterprise Linux？。

- Rocky Linux 9

  你可以使用官方的 Rocky Linux 9 AMIs 作为自定义的基础AMI。如果基础AMI版本AMI没有最新的内核，则您的自定义版本可能会失败。

  升级内核

  1. 从此处使用 rocky9 AMI ID 启动实例：https://rockylinux.org/cloud-images/
  2. 通过 ssh 登录实例并运行以下命令：

     ```
     sudo yum -y update
     ```

  3. 从实例创建镜像。您可以将此图像指定`ParentImage`为您的自定义图片AMI。

- Ubuntu 22.04

  Ubuntu 22.04 需要更安全的密钥，SSH并且默认情况下不支持RSA密钥。我们建议您改为生成和使用ED25519密钥。

> **Note**
>
> 你无法将 Ubuntu 22.04 更新到最新的内核，因为该内核没有FSx客户端。

中的 Slurm 版本 AWS PCS

SchedMD 通过新功能、优化和安全补丁不断增强 Slurm。SchedMD 定期发布新的主要版本，并计划在任何给定时间最多支持 3 个版本。 AWS PCS最初支持 Slurm 23.11。 AWS PCS旨在使用补丁版本自动更新 Slurm 控制器。

当 SchedMD 终止对特定主要版本的支持时， AWS PCS也将终止对该主要版本的支持。 AWS PCS如果 Slurm 主要版本接近其生命周期，则会提前发出通知，以帮助客户知道何时将其集群升级到更新的支持版本。

我们建议您使用最新支持的 Slurm 版本来部署集群，以访问最新的改进和改进。

有关 Slurm 版本的常见问题

AWS PCS支持 Slurm 版本需要多长时间？

AWS PCS遵循主要版本的 SchedMD 支持周期。 AWS PCS在任何给定时间最多支持 3 个主要版本。在 SchedMD 发布新的主要版本后，将 AWS PCS停用支持的最旧版本。 AWS PCS尽快发布 Slurm 的新主要版本，但在 SchedMD 的发布和发布之间可能会有延迟。 AWS PCS

什么时候会 AWS PCS通知我 Slurm 版本的 Support Life 终止 (EOSL)？

AWS PCS会在日期之前以预先确定的节奏多次通知您。EOSL

当 Slurm 版本临近时，我该怎么做？EOSL

您必须先更新 Slurm 版本EOSL，以帮助维护安全且受支持的环境。

如何更新我的集群以使用新的主版本的 Slurm？

要更新 Slurm 版本，必须创建一个新集群。您还必须升级到 Amazon 系统映像 (AMI) 中的等效 AWS PCS软件，并使用它为新集群创建计算节点组。

我的集群将如何获得新的 Slurm 补丁版本？

AWS PCS旨在自动应用补丁来解决 Slurm 常见漏洞和漏洞 ()。CVEs AWS PCS将补丁应用于在内部服务拥有的帐户中运行的集群控制器。要在中的EC2实例上安装补丁 AWS 账户，请更新您的计算节点组的，并更新计算节点组以使用更新后的补丁AMI。AMI有关更多信息，请参阅 定制 Amazon 机器映像 (AMIs) 适用于 AWS PCS。

> **Note**
>
> 当我们更新 Slurm 控制器时，它们不可用。正在运行的作业不受影响。当集群的控制器不可用时提交的作业将一直保留，直到控制器可用为止。

如果我不在截止日期之前更新 Slurm 怎么办？EOSL

AWS PCS旨在阻止具有不支持的 Slurm 版本的集群。您必须更新集群控制器的 Slurm 主版本和安装在计算节点组上的 AWS PCS软件。

支持多少个 Slurm 版本？ AWS PCS

AWS PCS在任何给定时间最多支持 3 个主要 Slurm 版本，包括当前版本和 2 个之前的主要版本。

我应该应用哪些 Slurm 版本更新？

我们强烈建议您在集群中的所有组件中使用相同的主要版本，并在最新补丁发布后立即安装它们。您的计算节点组必须使用与 Slurm 版本的集群控制器兼容的 Slurm 软件版本。AMIs你的 Slurm 主版本 AMIs必须与集群控制器上的 Slurm 主版本相比在 2 个版本之内。安装在AMI和集群中正在运行的EC2实例上的 Slurm 版本不能比集群控制器上的 Slurm 版本更新。要保持对集群的支持，AMIs必须使用支持的 AWS PCS软件版本。

如果我更新了 Slurm 主版本，但在我的AMI计算节点组中使用了较旧的 Slurm 软件，该怎么办？

必须将 AWS PCS软件更新到相同版本才能使用新的 Slurm 功能。要获得全面 AWS PCS支持，所有 Slurm 组件都必须使用支持的版本。总而言之：

- 当集群控制器及其中的所有组件（AWS PCS包） AWS 账户 都使用支持的版本时，我们能够提供全面支持。
- AWS PCS旨在在 Slurm 版本的控制器到达时停止集群。EOSL
- 如果您使用的是 Slurm 版本的组件EOSL， AWS 账户 则您的集群将不受支持。

我应该按什么顺序更新集群中的组件？

在使用较新版本的 Slurm 之前，必须更新集群控制器的 Slurm 版本。AMI您可以更新计算节点组以使用AMI。 AWS PCSAMI使用启动计算节点组中的新EC2实例。 AWS PCS不会更新正在运行的任务的现有EC2实例； AWS PCS旨在在这些实例的任务完成后将其终止。

是否为 Slurm 版本 AWS PCS提供扩展支持？

不是。 我们将提供有关扩展支持选项的详细信息，包括任何额外费用和所提供的具体支持范围。

AWS 并行计算服务中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 AWS 并行计算AWS 服务的合规性计划，请参阅按合规计划划分的范围内的合规性计划AWS 服务按合规计划。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 AWS PCS 时如何应用分担责任模型。以下主题向您介绍如何配置 AWS PCS 以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS PCS 资源。

主题

AWS 并行计算服务中的数据保护

分 AWS 担责任模型适用于 AWS 并行计算服务中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责

您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 使用 SSL/TLS 与资源通信。 AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您 AWS 服务 使用控制台、API 或与 AWS PCS 或其他人合作时 AWS SDKs。 AWS CLI在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

静态加密

使用、、PCS API 或创建 AWS 并行计算服务 (AWS PCS) 集群时 AWS Management Console AWS CLI， AWS 默认情况下会对静态数据启用加密 AWS SDKs。 AWS PCS 使用AWS 拥有的 KMS 密钥对静态数据进行加密。有关更多信息，请参阅《AWS KMS 开发人员指南》中的客户 AWS 密钥和密钥。您也可以使用客户管理的密钥。有关更多信息，请参阅 在 PCS 中使用加密 EBS 卷所必需的 KMS 密钥策略 AWS。

集群密钥存储在 Sec rets Manager 托管的 KMS 密钥中， AWS Secrets Manager 并使用 KMS 密钥进行加密。有关更多信息，请参阅 在中使用集群密钥 AWS PCS。

在 AWS PCS 集群中，以下数据处于静止状态：

- 调度器状态 — 它包括集群中正在运行的作业和已配置节点的数据。这是 Slurm 保留在你的`StateSaveLocation`定义中的数据。`slurm.conf`有关更多信息，请参阅 Slurm 文档StateSaveLocation中的描述。 AWS 任务完成后，PCS 会删除作业数据。
- 调度程序身份验证密钥 — AWS PCS 使用它来验证集群中的所有调度程序通信。

对于调度程序状态信息， AWS PCS 会在将数据和元数据写入文件系统之前自动对其进行加密。加密文件系统对静态数据使用行业标准的 AES-256 加密算法。

传输中加密

无论您使用 AWS Command Line Interface (AWS CLI) 还是，您与 AWS PCS API 的连接都使用签名版本 4 签名过程的 TLS 加密 AWS SDKs。有关更多信息，请参阅AWS Identity and Access Management 用户指南中的签署 AWS API 请求。 AWS 通过 API 管理您用于连接的安全证书的 IAM 策略的访问控制。

AWS PCS 使用 TLS 来连接其他 AWS 服务。

在 Slurm 集群中，调度器配置了`auth/slurm`身份验证插件，该插件可为所有调度程序通信提供身份验证。Slurm 不为其通信提供应用程序级别的加密，所有流经集群实例的数据都保留在 VP EC2 C 本地，因此，如果这些实例支持传输中的加密，则需要进行 VPC 加密。有关更多信息，请参阅《Amazon 弹性计算云用户指南》中的传输中加密。控制器（在服务帐户中配置）与您账户中的集群节点之间的通信是加密的。

密钥管理

AWS PCS 使用AWS 拥有的 KMS 密钥对数据进行加密。有关更多信息，请参阅《AWS KMS 开发人员指南》中的客户 AWS 密钥和密钥。您也可以使用客户管理的密钥。有关更多信息，请参阅 在 PCS 中使用加密 EBS 卷所必需的 KMS 密钥策略 AWS。

集群密钥存储在 Sec rets Manager 托管的 KMS 密钥中， AWS Secrets Manager 并使用 KMS 密钥进行加密。有关更多信息，请参阅 在中使用集群密钥 AWS PCS。

互联网络流量隐私

AWS 集群的 PCS 计算资源位于客户账户中的 1 个 VPC 内。因此，集群内的所有内部 AWS PCS 服务流量都留在 AWS 网络内，不会通过互联网传输。用户和 AWS PCS 节点之间的通信可以通过互联网传输，我们建议使用 SSH 或 Systems Manager 连接到节点。有关更多信息，请参阅什么是 AWS Systems Manager？ 在《AWS Systems Manager 用户指南》中。

您还可以使用以下产品将本地网络连接到 AWS：

- AWS Site-to-Site VPN。 有关更多信息，请参阅什么是 AWS Site-to-Site VPN？ 在《AWS Site-to-Site VPN 用户指南》中。
- 一个 AWS Direct Connect。有关更多信息，请参阅什么是 AWS Direct Connect？ 在《AWS Direct Connect 用户指南》中。

您可以访问 AWS PCS API 来执行服务的管理任务。您和您的用户访问 Slurm 端点端口，直接与调度程序进行交互。

加密 API 流量

要访问 AWS PCS API，客户端必须支持传输层安全 (TLS) 1.2 或更高版本。我们要求使用 TLS 1.2，建议使用 TLS 1.3。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。您还可以使用 AWS Security Token Service (AWS STS) 生成临时安全证书来签署请求。

加密数据流量

通过访问调度程序终端节点的受支持 EC2 实例以及内部的 ComputeNodeGroup 实例之间，可以对传输中的数据进行 AWS Cloud加密。有关更多信息，请参阅 传输中加密。

在 PCS 中使用加密 EBS 卷所必需的 KMS 密钥策略 AWS

AWS PCS 使用与服务相关的角色将权限委托给其他 AWS 服务人。 AWS PCS 服务相关角色是预定义的，包括 AWS PCS 代表您呼叫他人所需的权限。 AWS 服务 预定义的权限还包括对您的客户托管密钥的访问权限 AWS 托管式密钥 ，但不包括对您的客户托管密钥的访问权限。

本主题介绍在您为 Amazon EBS 加密指定客户托管密钥时，如何设置启动实例所需的密钥策略。

> **Note**
>
> AWS PCS 不需要额外的授权即可使用默认权限 AWS 托管式密钥 来保护您账户中的加密卷。

内容

- 示例 2：允许跨账户访问客户托管密钥的关键策略部分
- 在 AWS KMS 控制台中编辑密钥策略

概述

当 AWS PCS 启动实例时，您可以使用以下 AWS KMS keys 方法进行 Amazon EBS 加密：

- AWS 托管式密钥：您的账户中 Amazon EBS 创建、拥有和管理的加密密钥。这是新账户的默认加密密钥。除非您指定客户托管密钥， AWS 托管式密钥 否则 Amazon EBS 将使用进行加密。
- 客户托管密钥：您创建、拥有和管理的自定义加密密钥。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的创建 KMS 密钥。

> **Note**
>
> 密钥必须是对称的。Amazon EBS 不支持非对称的客户托管密钥。

在创建加密快照或指定加密卷的启动模板时，或者选择默认启用加密时，您可以配置客户托管密钥。

配置密钥策略

您的 KMS 密钥必须具有密钥策略，允许 AWS PCS 启动使用客户托管密钥加密的 Amazon EBS 卷的实例。

使用本页上的示例配置密钥策略，让 AWS PCS 可以访问您的客户托管密钥。您可以在创建密钥时或以后修改客户托管密钥的密钥策略。

密钥策略必须包含以下声明：

- 一种声明，允许`Principal`元素中指定的 IAM 身份直接使用客户托管密钥。它包括对密钥执行 AWS KMS `Encrypt`、`Decrypt`、`ReEncrypt*`、`GenerateDataKey*`、和`DescribeKey`操作的权限。
- 一种语句，允许`Principal`元素中指定的 IAM 身份使用该`CreateGrant`操作生成授权，这些授权将自己的一部分权限委托给与另一委托人 AWS KMS 或其他委托人集成的权限。 AWS 服务 这样，他们可以使用密钥代表您创建加密的资源。

在向密钥策略中添加新的政策声明时，请勿更改策略中的任何现有声明。

有关更多信息，请参阅：

- 命令参考中的 create-keyAWS CLI
- 《AWS CLI Command Reference》中的 put-key-policy。
- 在开发者指南中找到密钥 ID 和密钥 ARN AWS Key Management Service
- PCS 的服务相关角色 AWS
- 亚马逊 EBS 用户指南中的亚马逊 EBS 加密
- AWS Key Management Service在《AWS Key Management Service 开发者指南》中

示例 1：允许访问客户托管密钥的关键策略部分

将以下政策声明添加到客户托管密钥的密钥策略中。将示例 ARN 替换为服务相关角色的 ARN。AWSServiceRoleForPCS此示例策略授予 AWS PCS 服务相关角色 (AWSServiceRoleForPCS) 使用客户托管密钥的权限。

```
{
   "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
        }
     }
}
```

示例 2：允许跨账户访问客户托管密钥的关键策略部分

如果您在与 AWS PCS 集群不同的账户中创建客户托管密钥，则必须将授权与密钥策略结合使用，以允许跨账户访问该密钥。

授予对密钥的访问权限

1. 将以下政策声明添加到客户托管密钥的密钥策略中。将示例 ARN 替换为其他账户的 ARN。*111122223333*替换为要在中 AWS 账户 创建 AWS PCS 集群的的实际账户 ID。这将允许您向指定账户中的 IAM 用户或角色授予使用下面的 CLI 命令为密钥创建授权的权限。默认情况下，用户无权访问密钥。

```
{.
    "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. 使用您要在其中创建 AWS PCS 集群的账户创建授权，将相关权限委托给 AWS PCS 服务相关角色。的值`grantee-principal`是服务相关角色的 ARN。的值`key-id`是密钥的 ARN。

   以下示例 create-grant CLI 命令为账户`AWSServiceRoleForPCS`中指定的服务相关角色提供了在账户中使用客户托管密钥的*111122223333*权限。*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**
   >
   > 发出请求的用户必须拥有使用该`kms:CreateGrant`操作的权限。

   以下示例 IAM 策略允许账户中的 IAM 身份（用户或角色）*111122223333*为账户中的客户托管密钥创建授权*444455556666*。

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

有关为不同 AWS 账户中的 KMS 密钥创建授权的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 AWS KMS中的授权。

> **Important**
>
> 指定为被授权者委托人的服务相关角色名称必须是现有角色的名称。创建授权后，为确保该授权允许 AWS PCS 使用指定的 KMS 密钥，请勿删除和重新创建服务相关角色。

在 AWS KMS 控制台中编辑密钥策略

之前部分中的示例仅显示如何向密钥策略添加语句，这只是更改密钥策略的一种方法。更改密钥策略的最简单方法是使用 AWS KMS 控制台的默认密钥策略视图，并将 IAM 身份（用户或角色）设为相应密钥策略的关键用户之一。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的使用 AWS Management Console 默认视图。

> **Warning**
>
> 控制台的默认视图策略声明包括对客户托管密钥执行 AWS KMS Revoke操作的权限。如果您撤销授予您账户中客户托管密钥 AWS 账户 访问权限的授权，则该用户将 AWS 账户 失去对加密数据和密钥的访问权限。

AWS Parallel Computing Service 使用接口端点进行访问 (AWS PrivateLink)

您可以使用 AWS PrivateLink 在您的 VPC 和 AWS Parallel Computing Service (AWS PCS) 之间创建私有连接。您可以像在 VPC 中 AWS PCS 一样进行访问，无需使用互联网网关、NAT 设备、VPN 连接或 AWS Direct Connect 连接。VPC 中的实例不需要公有 IP 地址即可访问 AWS PCS。

您可以通过创建由 AWS PrivateLink提供支持的接口端点来建立此私有连接。我们将在您为接口端点启用的每个子网中创建一个端点网络接口。这些是请求者托管的网络接口，用作发往 AWS PCS的流量的入口点。

有关更多信息，请参阅AWS PrivateLink 指南 AWS PrivateLink中的AWS 服务 直通访问。

的注意事项 AWS PCS

在为设置接口终端节点之前 AWS PCS，请查看AWS PrivateLink 指南中的使用接口 VPC 终端节点访问 AWS 服务。

AWS PCS 支持通过接口端点调用其所有 API 操作。

如果您的 VPC 无法直接访问互联网，则必须配置 VPC 终端节点以使您的计算节点组实例能够调用 AWS PCS `RegisterComputeNodeGroupInstance` API 操作。

为创建接口终端节点 AWS PCS

您可以创建用于 AWS PCS 使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI) 的接口终端节点。有关更多信息，请参阅《AWS PrivateLink 指南》中的创建接口端点。

AWS PCS 使用以下服务名称创建接口终端节点：

```
com.amazonaws.region.pcs
```

*region*替换为 AWS 区域 用于创建终端节点的 ID，例如`us-east-1`。

如果为接口端点启用私有 DNS，则可使用其默认区域 DNS 名称向 AWS PCS 发出 API 请求。例如，`pcs.us-east-1.amazonaws.com`。

为 VPC 端点创建端点策略

端点策略是一种 IAM 资源，您可以将其附加到接口端点。默认终端节点策略允许 AWS PCS 通过接口终端节点进行完全访问。要控制允许 AWS PCS 从您的 VPC 访问权限，请将自定义终端节点策略附加到接口终端节点。

端点策略指定以下信息：

- 可执行操作的主体（AWS 账户、IAM 用户和 IAM 角色）。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅《AWS PrivateLink 指南》中的使用端点策略控制对服务的访问权限。

示例：用于 AWS PCS 操作的 VPC 终端节点策略

以下是自定义端点策略的示例。当您将此策略附加到接口终端节点时，它会向具有指定*cluster-id*权限的集群的所有委托人授予对所列 AWS PCS 操作的访问权限。*region*替换为集 AWS 区域 群的 ID，例如us-east-1。*account-id*替换为集群的 AWS 账户 编号。

```
{
    "Statement": [
            {
                "Action": [
                "pcs:CreateCluster",
                "pcs:ListClusters",
                "pcs:DeleteCluster",
                "pcs:GetCluster",
                ],
                "Effect": "Allow",
                "Principal": "*",
                "Resource": [
                    "arn:aws:pcs:region:account-id:cluster/cluster-id*"
                ]
            }
        ]
}
```

并 AWS 行计算服务的 Identity and Access 管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 AWS PCS 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 AWS PCS 中所做的工作。

服务用户-如果您使用 AWS PCS 服务完成工作，则您的管理员会为您提供所需的凭据和权限。当你使用更多的 AWS PCS 功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS PCS 中的某项功能，请参阅对 AWS 并行计算服务身份和访问进行故障排除。

服务管理员-如果您负责公司的 AWS PCS 资源，则可能拥有对 AWS PCS 的完全访问权限。您的工作是确定您的服务用户应访问哪些 AWS PCS 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解贵公司如何将 IAM 与 AWS PCS 结合使用，请参阅AWS 并行计算服务如何与 IAM 配合使用。

IAM 管理员 — 如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理 AWS PCS 的访问权限。要查看您可以在 IAM 中使用的基于身份的 AWS PCS 策略示例，请参阅。AWS 并行计算服务的基于身份的策略示例

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用建议的方法自行签署请求的更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

无论使用何种身份验证方法，您都可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的多重身份验证和《IAM 用户指南》中的在 IAM 中使用AWS 多重身份验证。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，请参阅 AWS IAM Identity Center 用户指南中的什么是 IAM Identity Center？。

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的用例，应在需要时更新访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可以拥有一个名为的群组，IAMAdmins并向该群组授予管理 IAM 资源的权限。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的 IAM 用户的使用案例。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。要在中临时担任 IAM 角色 AWS Management Console，您可以从用户切换到 IAM 角色（控制台）。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的代入角色的方法。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问：要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关用于联合身份验证的角色的信息，请参阅《IAM 用户指南》中的针对第三方身份提供商创建角色（联合身份验证）。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的权限集。
- 临时 IAM 用户权限：IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取：您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附

加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅 IAM 用户指南中的 IAM 中的跨账户资源访问。

- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在服务中拨打电话时，该服务通常会在 Amazon 中运行应用程序 EC2 或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种与服务相关联的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon 上运行的应用程序 EC2 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这比在 EC2 实例中存储访问密钥更可取。要为 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建一个附加到该实例的实例配置文件。实例配置文件包含该角色，并允许在 EC2 实例上运行的程序获得临时证书。有关更多信息，请参阅 IAM 用户指南中的使用 IAM 角色向在 A mazon EC2 实例上运行的应用程序授予权限。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间做出选择，请参阅《IAM 用户指南》中的在托管式策略和内联策略之间做出选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACLs)

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持的服务示例 ACLs。 AWS WAF要了解更多信息 ACLs，请参阅《亚马逊简单存储服务开发者指南》中的访问控制列表 (ACL) 概述。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界

的交集。在 Principal 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅IAM 用户指南中的 IAM 实体的权限边界。

- 服务控制策略 (SCPs)- SCPs 是指定组织或组织单位 (OU) 的最大权限的 JSON 策略 AWS Organizations。 AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果您启用组织中的所有功能，则可以将服务控制策略 (SCPs) 应用于您的任何或所有帐户。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organization SCPs s 和的更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。
- 资源控制策略 (RCPs) — RCPs 是 JSON 策略，您可以使用它来设置账户中资源的最大可用权限，而无需更新附加到您拥有的每个资源的 IAM 策略。RCP 限制成员账户中资源的权限，并可能影响身份（包括身份）的有效权限 AWS 账户根用户，无论这些身份是否属于您的组织。有关 Organizations 的更多信息 RCPs，包括 AWS 服务 该支持的列表 RCPs，请参阅《AWS Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略：会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

AWS 并行计算服务如何与 IAM 配合使用

在使用 IAM 管理对 AWS PCS 的访问权限之前，请先了解有哪些 IAM 功能可用于 AWS PCS。

您可以与 AWS 并行计算服务一起使用的 IAM 功能

IAM 特征	AWS 个人电脑支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是

IAM 特征	AWS 个人电脑支持
ACLs	否
ABAC（策略中的标签）	是
临时凭证	是
主体权限	是
服务角色	否
服务相关角色	是

要全面了解 AWS PCS 和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

PCS 基于身份的策略 AWS

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

PCS 基于身份的策略示例 AWS

要查看 AWS PCS 基于身份的策略的示例，请参阅。AWS 并行计算服务的基于身份的策略示例

PCS 中 AWS 基于资源的策略

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件

下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

AWS PCS 的政策措施

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 AWS PCS 操作列表，请参阅《服务授权参考》中的 “ AWS 并行计算服务定义的操作”。

AWS PCS 中的策略操作在操作前使用以下前缀：

```
pcs
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "pcs:action1",
      "pcs:action2"
         ]
```

AWS PCS 的策略资源

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

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

要查看 AWS PCS 资源类型及其列表 ARNs，请参阅《服务授权参考》中的“ AWS 并行计算服务定义的资源”。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅AWS 并行计算服务定义的操作。

要查看 AWS PCS 基于身份的策略的示例，请参阅。AWS 并行计算服务的基于身份的策略示例

AWS PCS 的策略条件密钥

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 策略元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 AWS PCS 条件密钥列表，请参阅《服务授权参考》中的“ AWS 并行计算服务的条件密钥”。要了解可以使用条件键的操作和资源，请参阅AWS 并行计算服务定义的操作。

要查看 AWS PCS 基于身份的策略的示例，请参阅。AWS 并行计算服务的基于身份的策略示例

ACLs 在 AWS PCS 中

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

带 PCS 的 ABA AWS C

支持 ABAC（策略中的标签）：是

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）和许多 AWS 资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

在 AWS PCS 中使用临时证书

支持临时凭证：是

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的AWS 服务 与 IA M 配合使用的信息。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。 AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

PCS 的跨服务主体 AWS 权限

支持转发访问会话（FAS）：是

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

AWS PCS 的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会中断 AWS PCS 的功能。仅当 AWS PCS 提供相关指导时才编辑服务角色。

PCS 的服务相关角色 AWS

支持服务相关角色：是

服务相关角色是一种与服务相关联的 AWS 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理 AWS PCS 服务相关角色的详细信息，请参阅PCS 的服务相关角色 AWS。

AWS 并行计算服务的基于身份的策略示例

默认情况下，用户和角色无权创建或修改 AWS PCS 资源。他们也无法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源

执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 AWS PCS 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》中的 “AWS 并行计算服务的操作、资源和条件密钥”。 ARNs

主题

- 策略最佳实践
- 使用 AWS PCS 控制台
- 允许用户查看他们自己的权限

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS PCS 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管式策略或工作职能的AWS 托管式策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

使用 AWS PCS 控制台

要访问 AWS 并行计算服务控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您的 AWS PCS 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

有关使用 AWS PCS 控制台所需的最低权限的更多信息，请参阅AWS PCS 的最低权限。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "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": "*"
        }
    ]
}
```

AWSAWS 并行计算服务的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管策略。

AWS 托管策略： AWSPCSServiceRolePolicy

您无法附加 AWSPCSServiceRolePolicy 到您的 IAM 实体。此策略附加到服务相关角色，允许 AWS PCS 代表您执行操作。有关更多信息，请参阅 PCS 的服务相关角色 AWS。

权限详细信息

该策略包含以下权限。

- `ec2`— 允许 AWS PCS 创建和管理 Amazon EC2 资源。
- `iam`— 允许 AWS PCS 为亚马逊 EC2 舰队创建服务相关角色并将该角色传递给亚马逊 EC2。
- `cloudwatch`— 允许 AWS PCS 向亚马逊发布服务指标 CloudWatch。
- `secretsmanager`— 允许 AWS PCS 管理 AWS PCS 群集资源的密钥。

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "PermissionsToCreatePCSNetworkInterfaces",
      "Effect" : "Allow",
      "Action" : [
        "ec2:CreateNetworkInterface"
      ],
      "Resource" : "arn:aws:ec2:*:*:network-interface/*",
      "Condition" : {
        "Null" : {
          "aws:RequestTag/AWSPCSManaged" : "false"
```

```
        }
      }
    },
    {
      "Sid" : "PermissionsToCreatePCSNetworkInterfacesInSubnet",
      "Effect" : "Allow",
      "Action" : [
        "ec2:CreateNetworkInterface"
      ],
      "Resource" : [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid" : "PermissionsToManagePCSNetworkInterfaces",
      "Effect" : "Allow",
      "Action" : [
        "ec2:DeleteNetworkInterface",
        "ec2:CreateNetworkInterfacePermission"
      ],
      "Resource" : "arn:aws:ec2:*:*:network-interface/*",
      "Condition" : {
        "Null" : {
          "aws:ResourceTag/AWSPCSManaged" : "false"
        }
      }
    },
    {
      "Sid" : "PermissionsToDescribePCSResources",
      "Effect" : "Allow",
      "Action" : [
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeImages",
```

```
        "ec2:DescribeImageAttribute"
      ],
      "Resource" : "*"
    },
    {
      "Sid" : "PermissionsToCreatePCSLaunchTemplates",
      "Effect" : "Allow",
      "Action" : [
        "ec2:CreateLaunchTemplate"
      ],
      "Resource" : "arn:aws:ec2:*:*:launch-template/*",
      "Condition" : {
        "Null" : {
          "aws:RequestTag/AWSPCSManaged" : "false"
        }
      }
    },
    {
      "Sid" : "PermissionsToManagePCSLaunchTemplates",
      "Effect" : "Allow",
      "Action" : [
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions",
        "ec2:CreateLaunchTemplateVersion"
      ],
      "Resource" : "arn:aws:ec2:*:*:launch-template/*",
      "Condition" : {
        "Null" : {
          "aws:ResourceTag/AWSPCSManaged" : "false"
        }
      }
    },
    {
      "Sid" : "PermissionsToTerminatePCSManagedInstances",
      "Effect" : "Allow",
      "Action" : [
        "ec2:TerminateInstances"
      ],
      "Resource" : "arn:aws:ec2:*:*:instance/*",
      "Condition" : {
        "Null" : {
          "aws:ResourceTag/AWSPCSManaged" : "false"
        }
      }
```

```
    },
    {
      "Sid" : "PermissionsToPassRoleToEC2",
      "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" : "PermissionsToControlClusterInstanceAttributes",
      "Effect" : "Allow",
      "Action" : [
        "ec2:RunInstances",
        "ec2:CreateFleet"
      ],
      "Resource" : [
        "arn:aws:ec2:*::image/*",
        "arn:aws:ec2:*::snapshot/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*",
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:key-pair/*",
        "arn:aws:ec2:*:*:launch-template/*",
        "arn:aws:ec2:*:*:placement-group/*",
        "arn:aws:ec2:*:*:capacity-reservation/*",
        "arn:aws:resource-groups:*:*:group/*",
        "arn:aws:ec2:*:*:fleet/*",
        "arn:aws:ec2:*:*:spot-instances-request/*"
      ]
    },
    {
      "Sid" : "PermissionsToProvisionClusterInstances",
```

```
      "Effect" : "Allow",
      "Action" : [
        "ec2:RunInstances",
        "ec2:CreateFleet"
      ],
      "Resource" : [
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition" : {
        "Null" : {
          "aws:RequestTag/AWSPCSManaged" : "false"
        }
      }
    },
    {
      "Sid" : "PermissionsToTagPCSResources",
      "Effect" : "Allow",
      "Action" : [
        "ec2:CreateTags"
      ],
      "Resource" : [
        "*"
      ],
      "Condition" : {
        "StringEquals" : {
          "ec2:CreateAction" : [
            "RunInstances",
            "CreateLaunchTemplate",
            "CreateFleet",
            "CreateNetworkInterface"
          ]
        }
      }
    },
    {
      "Sid" : "PermissionsToPublishMetrics",
      "Effect" : "Allow",
      "Action" : "cloudwatch:PutMetricData",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "cloudwatch:namespace" : "AWS/PCS"
        }
      }
```

```
    },
    {
      "Sid" : "PermissionsToManageSecret",
      "Effect" : "Allow",
      "Action" : [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "secretsmanager:UpdateSecretVersionStage",
        "secretsmanager:DeleteSecret"
      ],
      "Resource" : "arn:aws:secretsmanager:*:*:secret:pcs!*",
      "Condition" : {
        "StringEquals" : {
          "secretsmanager:ResourceTag/aws:secretsmanager:owningService" : "pcs",
          "aws:ResourceAccount" : "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

AWSAWS 托管策略的 PCS 更新

查看自该服务开始跟踪这些更改以来对 AWS PCS AWS 托管策略的更新的详细信息。要获得有关此页面变更的自动提醒，请订阅 AWS PCS 文档历史记录页面上的 RSS 提要。

更改	描述	日期
更新了本文档中的 JSON	更正了本文档中的 JSON 以包含在内`"arn:aws:ec2:*:*:spot-instances-request/*"` 。	2024 年 9 月 5 日
AWS PCS 开始追踪变更	AWS PCS 开始跟踪其 AWS 托管策略的更改。	2024 年 8 月 28 日

PCS 的服务相关角色 AWS

AWS 并行计算服务使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特的 IAM 角色，直接关联到 AWS PCS。服务相关角色由 AWS PCS 预定义，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可以更轻松地设置 AWS PCS，因为您不必手动添加必要的权限。 AWS PCS 定义其服务相关角色的权限，除非另有定义，否则只有 AWS PCS 可以担任其角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其它 IAM 实体。

只有在首先删除服务相关角色的相关资源后，才能删除该角色。这样可以保护您的 AWS PCS 资源，因为您不会意外删除访问这些资源的权限。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的AWS 服务，并在服务相关角色列中查找标有 “是” 的服务。选择是和链接，查看该服务的服务相关角色文档。

PCS 的服务相关角色权限 AWS

AWS PCS 使用名为 PC AWSServiceRoleForS 的服务相关角色——允许 AWS PCS 管理亚马逊 EC2 资源。

AWSServiceRoleForPCS 服务相关角色信任以下服务来代入该角色：

- `pcs.amazonaws.com`

名为的角色权限策略AWSPCSServiceRolePolicy允许 AWS PCS 完成对特定资源的操作。

您必须配置使用户、组或角色能够创建、编辑或删除服务相关角色的权限。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 PCS 创建服务相关角色 AWS

您无需手动创建服务相关角色。 AWS 创建集群时，PCS 会为您创建一个服务相关角色。

编辑 PCS 的服务相关角色 AWS

AWS PCS 不允许您编辑 AWSService RoleFor PCS 服务相关角色。创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 PCS 的服务相关角色 AWS

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。但是，必须先清除服务相关角色的资源，然后才能手动删除它。

> **Note**
>
> 如果您尝试删除资源时 AWS PCS 服务正在使用该角色，则删除可能会失败。如果发生这种情况，请等待几分钟后重试。

移除 AWS PCS 使用的 AWSService RoleFor PCS 资源

您必须删除所有集群才能删除 AWSService RoleFor PCS 服务相关角色。有关更多信息，请参阅删除集群。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除 AWSService RoleFor PCS 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

AWS PCS 服务相关角色支持的区域

AWS PCS 支持在提供服务的所有地区使用服务相关角色。有关更多信息，请参阅AWS 区域和端点。

适用于 AWS PCS 的 Amazon EC2 Spot 角色

如果要创建使用 Spot 作为购买选项的 AWS PCS 计算节点组，则还必须具有AWSServiceRoleForEC2竞价服务相关角色。 AWS 账户您可以使用以下 AWS CLI 命令来创建角色。有关更多信息，请参阅《AWS Identity and Access Management 用户指南》中的创建服务相关角色和创建向 AWS 服务委派权限的角色。

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

> **Note**
>
> 如果您 AWS 账户 已经拥有 `AWSServiceRoleForEC2Spot` IAM 角色，则会收到以下错误。

```
 An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
operation: Service role name AWSServiceRoleForEC2Spot has been taken in this
account, please try a different suffix.
```

AWS PCS 的最低权限

本节介绍了 IAM 身份（用户、群组或角色）使用该服务所需的最低 IAM 权限。

目录

使用 API 操作的最低权限

API 操作	最小权限	控制台的其他权限
CreateCluster	`ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:GetSecurityGroupsForVpc, iam:CreateServiceLinkedRole, secretsmanager:CreateSecret, secretsmanager:TagResource, pcs:CreateCluster`	
ListClusters	`pcs:ListClusters`	

API 操作	最小权限	控制台的其他权限
GetCluster	`pcs:GetCluster`	`ec2:DescribeSubnets`
DeleteCluster	`pcs:DeleteCluster`	
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`

API 操作	最小权限	控制台的其他权限
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`	
UpdateQueue	`pcs:UpdateQueue`	`pcs:ListComputeNodeGroups, pcs:GetQueue`

API 操作	最小权限	控制台的其他权限
DeleteQueue	`pcs:DeleteQueue`	

使用标签的最低权限

在 AWS PCS 中对资源使用标签需要以下权限。

```
pcs:ListTagsForResource,
pcs:TagResource,
pcs:UntagResource
```

支持日志的最低权限

AWS PCS 将日志数据发送到 Amazon CloudWatch 日志（CloudWatch 日志）。您必须确保您的身份具有使用 CloudWatch 日志的最低权限。有关更多信息，请参阅 Amazon Logs 用户指南中的管理 CloudWatch CloudWatch 日志资源访问权限概述。

有关服务向日志发送日志所需的权限的信息，请参阅 Amazon CloudWatch Lo g CloudWatch s 用户指南中的启用 AWS 服务日志记录。

服务管理员的最低权限

以下 IAM 策略指定了 IAM 身份（用户、群组或角色）配置和管理 AWS PCS 服务所需的最低权限。

> **Note**
>
> 不配置和管理服务的用户不需要这些权限。仅运行作业的用户使用安全外壳 (SSH) 连接到集群。 AWS Identity and Access Management (IAM) 不处理 SSH 的身份验证或授权。

```
{
  "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"
      ],
      "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"
      ],
      "Resource": "*"
    },
    {
       "Sid": "ServiceLogsDelivery",
       "Effect": "Allow",
       "Action": [
         "pcs:AllowVendedLogDeliveryForResource",
         "logs:PutDeliverySource",
         "logs:PutDeliveryDestination",
         "logs:CreateDelivery"
       ],
       "Resource": "*"
    }
  ]
}
```

AWS 并行计算服务的 IAM 实例配置文件

在 EC2 实例上运行的应用程序必须在其发出的任何 AWS API 请求中包含 AWS 证书。我们建议您使用 IAM 角色来管理 EC2 实例上的临时证书。您可以定义实例配置文件来执行此操作，并将其附加到您的实例。有关更多信息，请参阅《亚马逊弹性计算云用户指南》 EC2中的 Amazon IAM 角色。

> **Note**
>
> 当您使用为 Amazon 创建 IAM 角色时 EC2，控制台会自动创建实例配置文件并将其命名为 IAM 角色。 AWS Management Console 如果您使用 AWS CLI、 AWS API 操作或 AWS 软件开发工具包创建 IAM 角色，则可以将实例配置文件作为单独的操作创建。有关更多信息，请参阅 Amazon 弹性计算云用户指南中的实例配置文件。

创建计算节点组时，必须指定实例配置文件的 Amazon 资源名称 (ARN)。您可以为部分或所有计算节点组选择不同的实例配置文件。

实例配置文件要求

实例配置文件 ARN

ARN 的 IAM 角色名称部分必须以以下开头AWSPCS或在其路径/aws-pcs/中包含：

- `arn:aws:iam::*:instance-profile/AWSPCS-example-role-1` 和
- `arn:aws:iam::*:instance-profile/aws-pcs/example-role-2`.

> **Note**
>
> 如果您使用 AWS CLI，请为提供一个`--path`值`iam create-instance-profile`以包含`/aws-pcs/`在 ARN 路径中。例如：
>
> ```
> aws iam create-instance-profile --path /aws-pcs/ --instance-profile-name
>  example-role-2
> ```

权限

AWS PCS 的实例配置文件必须至少包含以下策略。它允许计算节点在开始运行时通知 AWS PCS 服务。

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

其他政策

您可以考虑将托管策略添加到实例配置文件中。例如：

- AmazonS3 ReadOnlyAccess 提供对所有 S3 存储桶的只读访问权限。
- 亚马逊SSMManagedInstanceCore支持 AWS Systems Manager 服务的核心功能，例如直接从亚马逊管理控制台进行远程访问。
- CloudWatchAgentServerPolicy包含 AmazonCloudWatchAgent 在服务器上使用所需的权限。

您也可以加入自己的 IAM 策略来支持您的特定使用案例。

创建实例配置文件

您可以直接从 Amazon EC2 控制台创建实例配置文件。有关更多信息，请参阅AWS Identity and Access Management 用户指南中的使用实例配置文件。

对 AWS 并行计算服务身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS PCS 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 AWS PCS 中执行操作
- 我无权执行 iam：PassRole
- 我想允许我以外的人 AWS 账户 访问我的 AWS PCS 资源

我无权在 AWS PCS 中执行操作

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 `mateojackson` IAM 用户尝试使用控制台查看有关虚构 *`my-example-widget`* 资源的详细信息，但不拥有虚构 `pcs:`*`GetWidget`* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 pcs:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 `mateojackson` 用户的策略，以允许使用 `pcs:`*`GetWidget`* 操作访问 *`my-example-widget`* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到错误消息，提示您无权执行`iam:PassRole`操作，则必须更新您的策略以允许您将角色传递给 AWS PCS。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为的 IAM 用户marymajor尝试使用控制台在 AWS PCS 中执行操作时，会出现以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 AWS PCS 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 AWS PCS 是否支持这些功能，请参阅AWS 并行计算服务如何与 IAM 配合使用。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

AWS 并行计算服务的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- Security Compliance & Governance：这些解决方案实施指南讨论了架构考虑因素，并提供了部署安全性和合规性功能的步骤。
- 在 A@@ mazon Web Services 上构建 HIPAA 安全与合规性 — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅符合 HIPAA 要求的服务参考。

- AWS 合AWS 规资源 — 此工作簿和指南集合可能适用于您的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控制措施评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控制措施的列表，请参阅 Security Hub 控制措施参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务检测您的工作负载、容器和数据面临的潜在威胁。 GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

AWS 并行计算服务中的弹性

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。 AWS 区域 提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础结构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

AWS 并行计算服务中的基础设施安全

作为一项托管服务，AWS 并行计算服务受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 AWS PCS。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

当 AWS PCS 创建集群时，该服务将在服务拥有的账户中启动 Slurm 控制器，该账户与您账户中的计算节点分开。为了桥接控制器和计算节点之间的通信，AWS PCS 在您的 VPC 中创建一个跨账户弹性网络接口 (ENI)。Slurm 控制器使用 ENI 来管理不同计算节点并与之通信 AWS 账户，维护资源的安全和隔离，同时促进高效的 HPC 和 AI/ML 操作。

并 AWS 行计算服务中的漏洞分析和管理

配置和 IT 控制是您 AWS 和您的共同责任。有关更多信息，请参阅责任AWS 共担模型。 AWS 处理服务帐户中底层基础设施的基本安全任务，例如在控制器实例上修补操作系统、配置防火墙和 AWS 基础设施灾难恢复。这些流程已通过相应第三方审核和认证。有关更多详细信息，请参阅安全性、身份和合规性最佳实践。

> **Note**
>
> 当我们更新 Slurm 控制器时，它们不可用。正在运行的作业不受影响。当集群的控制器不可用时提交的作业将一直保留，直到控制器可用为止。

您应对以下基础架构的安全性负责 AWS 账户：

- 维护您的代码，包括更新和安全补丁。
- 修补和更新计算节点组的 Amazon 系统映像 (AMI) 中的操作系统，更新计算节点组以使用更新后的 AMI。

- 更新调度程序以使其保持在支持的版本内。更新计算节点组的 AMI，更新计算节点组以使用更新后的 AMI。
- 对用户客户端与其连接的节点之间的通信进行身份验证和加密。

有关更新计算节点组的 AMI 的更多信息，请参阅Amazon 机器映像 (AMIs) 适用于 AWS PCS。

防止跨服务混淆座席

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况， AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在资源策略中使用aws:SourceArn和aws:SourceAccount全局条件上下文密钥来限制 AWS 并行计算服务 (AWS PCS) 向该资源提供的其他服务的权限。如果您只希望将一个资源与跨服务访问相关联，请使用 aws:SourceArn。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 aws:SourceAccount。

防范混淆代理问题最有效的方法是使用 aws:SourceArn 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符字符 (*) 的 aws:SourceArn 全局上下文条件键。例如，arn:aws:*servicename*:*:*123456789012*:*。

如果 aws:SourceArn 值不包含账户 ID，例如 Amazon S3 存储桶 ARN，您必须使用两个全局条件上下文键来限制权限。

的值aws:SourceArn必须是集群 ARN。

以下示例显示了如何在 AWS PCS 中使用aws:SourceArn和aws:SourceAccount全局条件上下文键来防止出现混淆的副手问题。

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

作为计算节点组一部分预置的 Amazon EC2 实例的 IAM 角色

AWS PCS 会自动为集群中每个已配置的计算节点组编排 Amazon EC2 容量。创建计算节点组时，用户必须通过`iamInstanceProfileArn`字段提供 IAM 实例配置文件。实例配置文件指定了与预配置 EC2 实例关联的权限。 AWS PCS 接受任何具有`AWSPCS`角色名称前缀或`/aws-pcs/`角色路径一部分的角色。创建或更新计算节点组的 IAM 身份（用户或角色）需要`iam:PassRole`获得权限。当用户调用`CreateComputeNodeGroup`或 UpdateComputeNodeGroup API 操作时， AWS PCS 会检查是否允许该用户执行`iam:PassRole`操作。

下面的策略示例授予仅传递名称以 AWSPCS 开头的 IAM 角色的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/AWSPCS*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        }
    ]
```

```
}
```

AWS 并行计算服务的安全最佳实践

本节介绍特定于 AWS 并行计算服务 (AWS PCS) 的安全最佳实践。要详细了解中的安全最佳实践 AWS，请参阅安全、身份和合规性最佳实践。

与 AMI 相关的安全性

- 不要将 AWS PCS 示例 AMIs 用于生产工作负载。该样本 AMIs 不受支持，仅用于测试。
- 定期更新计算节点组的 AMI 中的操作系统和软件，以缓解漏洞。
- 仅使用从官方 AWS 来源下载的经过身份验证的官方 AWS PCS 软件包。
- 定期更新 AMI 中计算节点组的 AWS PCS 包，并更新计算节点以使用更新后的 AMI。考虑将此过程自动化，以最大限度地减少漏洞。

有关更多信息，请参阅 定制 Amazon 机器映像 (AMIs) 适用于 AWS PCS。

Slurm 工作负载管理器安全

- 实施访问控制和网络限制，以保护 Slurm 控制和计算节点。仅允许受信任的用户和系统提交作业和访问 Slurm 管理命令。
- 使用 Slurm 的内置安全功能（例如 Slurm 身份验证）来确保提交的工作和通信经过身份验证。
- 更新 Slurm 版本以保持流畅的操作和集群支持。

> **Important**
>
> 任何使用已达到支持寿命 (EOSL) 的 Slurm 版本的集群都将立即停止。使用用户指南页面顶部的链接订阅 AWS PCS 文档 RSS 提要，以便在 Slurm 版本接近 EOSL 时收到通知。

有关更多信息，请参阅 中的 Slurm 版本 AWS PCS。

监控和日志记录

- 使用 Amazon CloudWatch Logs and AWS CloudTrail 来监控和记录集群中的操作，以及 AWS 账户。使用这些数据进行故障排除和审计。

网络安全

- 将 AWS PCS 集群部署在单独的 VPC 中，将您的 HPC 环境与其他网络流量隔离开来。
- 使用安全组和网络访问控制列表 (ACLs) 来控制 AWS PCS 实例和子网的入站和出站流量。
- 使用 AWS PrivateLink 我们的 VPC 终端节点将网络流量保持在您的集群和 AWS 网络内的其他 AWS 服务之间。有关更多信息，请参阅 AWS Parallel Computing Service 使用接口端点进行访问 (AWS PrivateLink)。

记录和监控 AWS PCS

监控是维护和其他AWS资源的可靠性、可用性和性能的重要组成部分。 AWS PCSAWS提供以下监控工具 AWS PCS，供您监视、报告问题并在适当时自动采取措施：

- Amazon 会实时 CloudWatch监控您的 AWS 资源和您运行 AWS 的应用程序。您可以收集和跟踪指标，创建自定义的控制平面，以及 设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。例如，您可以 CloudWatch 跟踪您的 Amazon EC2 实例的CPU使用情况或其他指标，并在需要时自动启动新实例。有关更多信息，请参阅 Amazon CloudWatch 用户指南。
- Amazon Lo CloudWatch gs 使您能够监控、存储和访问来自亚马逊EC2实例和其他来源的日志文件。 CloudTrail CloudWatch 日志可以监视日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 Amazon CloudWatch 日志用户指南。
- AWS CloudTrail捕获由您的账户或代表您的 AWS 账户发出的API调用和相关事件，并将日志文件传输到您指定的 Amazon S3 存储桶。您可以识别哪些用户和帐户拨打了电话 AWS、发出呼叫的源 IP 地址以及呼叫发生的时间。有关更多信息，请参阅 用户指南。AWS CloudTrail

AWS PCS调度程序日志

您可以配置 AWS PCS为将详细的日志数据从集群计划程序发送到亚马逊 CloudWatch 日志、亚马逊简单存储服务 (Amazon S3) Service 和 Amazon Data Firehose。这可以帮助进行监控和故障排除。您可以使用 AWS PCS控制台设置 AWS PCS调度程序日志，也可以使用或以编程方式设置日志。 AWS CLI SDK

目录

先决条件

用于管理AWSPCS集群的IAM委托人必须允许pcs:AllowVendedLogDeliveryForResource。以下是启用它的AWSIAM策略示例。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PcsAllowVendedLogsDelivery",
            "Effect": "Allow",
            "Action": ["pcs:AllowVendedLogDeliveryForResource"],
            "Resource": [
                "arn:aws:pcs:::cluster/*"
            ]
        }
    ]
}
```

使用控制台设置调度程序日志 AWS PCS

要在控制台中设置AWSPCS调度程序日志，请执行以下步骤：

1. 打开控制AWS PCS台。
2. 选择 Clu st ers，然后导航到要启用日志记录的 AWS PCS集群的详细信息页面。
3. 选择 Logs (日志)。
4. 在日志传送下-调度器日志-可选
   a. 最多添加三个日志传送目的地。选项包括 CloudWatch 日志、亚马逊 S3 或 Firehose。
   b. 选择 “更新日志传送”。

您可以通过重新访问此页面来重新配置、添加或删除日志传送。

使用设置调度程序日志 AWS CLI

为此，您需要至少一个交付目标、一个交付源（PCS集群）和一个交付（一种将源与目标连接起来的关系）。

创建配送目的地

要接收来自AWSPCS集群的调度程序日志，您至少需要一个传送目标。您可以在《 CloudWatch API用户指南》 PutDeliveryDestination部分中了解有关此主题的更多信息。

要使用创建配送目的地 AWS CLI

- 使用以下命令创建目的地。在运行命令之前，进行以下替换：
  - Replace（替换） *region-code* 以及你将在 AWS 区域 哪里创建目的地。这通常与 AWS PCS集群的部署区域相同。
  - Replace（替换） *pcs-logs-destination* 用你的首选名字。对于您账户中的所有配送目的地，它必须是唯一的。
  - Replace（替换） *resource-arn* ARN用于日志中的 CloudWatch 现有日志组、S3 存储桶或 Firehose 中的传输流。示例包括：
    - CloudWatch 日志组

      ```
      arn:aws:logs:region-code:account-id:log-group:/log-group-name:*
      ```

    - S3 bucket

      ```
      arn:aws:s3:::bucket-name
      ```

    - Firehose 传送直播

      ```
      arn:aws:firehose:region-code:account-id:deliverystream/stream-name
      ```

```
aws logs put-delivery-destination --region region-code \
    --name pcs-logs-destination \
    --delivery-destination-configuration destinationResourceArn=resource-arn
```

请记下ARN新配送目的地的，因为您将需要它来配置配送。

启用集 AWS PCS群作为交付源

要从中收集调度程序日志 AWSPCS，请将集群配置为交付源。有关更多信息，请参阅 Amazon CloudWatch 日志API参考PutDeliverySource中的。

要将集群配置为交付源，请使用 AWS CLI

- 使用以下命令启用集群中的日志传输。在运行命令之前，进行以下替换：
  - Replace（替换）*region-code* 以及您的 AWS 区域 集群的部署位置。
  - Replace（替换）*cluster-logs-source-name* 用这个来源的名字。对于您的所有交付来源，它必须是唯一的 AWS 账户。考虑合并集 AWS PCS群的名称或 ID。
  - Replace（替换）*cluster-arn* 使用ARN适用于您的 AWS PCS集群

```
aws logs put-delivery-source \
    --region region-code \
    --name cluster-logs-source-name \
    --resource-arn cluster-arn \
    --log-type PCS_SCHEDULER_LOGS
```

将集群交付源连接到传送目标

要使调度程序日志数据从集群流向目标，必须配置连接集群的传输。有关更多信息，请参阅 Amazon CloudWatch 日志API参考CreateDelivery中的。

要使用创建配送 AWS CLI

- 使用以下命令创建交付。在运行命令之前，进行以下替换：
  - Replace（替换）*region-code* 以及您的来源和目的地 AWS 区域 所在的位置。
  - Replace（替换）*cluster-logs-source-name* 上面写上你的配送来源的名称。
  - Replace（替换）*destination-arn* ARN从您要将日志传送到的传送目的地。

```
aws logs create-delivery \
    --region region-code \
    --delivery-source-name cluster-logs-source \
    --delivery-destination-arn destination-arn
```

调度器日志流路径和名称

AWSPCS计划程序日志的路径和名称取决于目标类型。

- CloudWatch 日志
  - CloudWatch 日志流遵循此命名约定。

```
AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
```

Example

```
AWSLogs/PCS/abcdef0123/slurmctld_24.05.log
```

- S3 bucket
  - S3 存储桶输出路径遵循以下命名约定：

```
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.
```

  - S3 对象名称遵循以下约定：

```
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
```

AWS PCS调度器日志记录示例

AWSPCS调度程序日志是结构化的。除了 Slurm 控制器进程发出的日志消息外，它们还包括集群标识符、调度器类型、主要版本和补丁版本等字段。下面是一个例子。

```
{
    "resource_id": "s3431v9rx2",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1721230979,
    "log_level": "info",
    "log_name": "slurmctld",
    "scheduler_type": "slurm",
    "scheduler_major_version": "23.11",
```

```
    "scheduler_patch_version": "8",
    "node_type": "controller_primary",
    "message": "[2024-07-17T15:42:58.614+00:00] Running as primary controller\n"
}
```

使用 Amazon 监控 AWS 并行计算服务 CloudWatch

Amaz CloudWatch on 通过定期从集群收集指标，监控您的并 AWS 行计算服务 (AWS PCS) 集群的运行状况和性能。这些指标会被保留，使您可以访问历史数据并深入了解集群在一段时间内的性能。

CloudWatch 还允许您监控启动的EC2实例 AWS PCS，以满足您的扩展要求。虽然您可以检查正在运行的实例的日志，但 CloudWatch 指标和日志数据通常会在实例终止后删除。但是，您可以使用EC2启动模板在实例上配置 CloudWatch 代理，使其即使在实例终止后也能保留指标和日志，从而实现长期监控和分析。

浏览本节的主题，详细了解如何 AWS PCS使用进行监控 CloudWatch。

主题

- 使用监控 AWS PCS指标 CloudWatch
- 使用亚马逊监控 AWS PCS 实例 CloudWatch

使用监控 AWS PCS指标 CloudWatch

您可以使用 Amazon 监控 AWS PCS集群运行状况 CloudWatch，Amazon 会从您的集群中收集数据并将其转换为近乎实时的指标。这些统计数据会保留 15 个月，因此您可以访问历史信息并更好地了解集群的性能。集群指标以 1 分钟 CloudWatch 为周期发送到。有关的更多信息 CloudWatch，请参阅 Amazon 是什么 CloudWatch？ 在《亚马逊 CloudWatch 用户指南》中。

AWS PCS将以下指标发布到中的 AWS/PCS命名空间 CloudWatch。它们只有一个维度，`ClusterId`。

名称	描述	单位
ActualCapacity	IdleCapacity + UtilizedCapacity	计数
CapacityUtilization	UtilizedCapacity / ActualCapacity	计数

名称	描述	单位
DesiredCapacity	ActualCapacity + PendingCapacity	计数
IdleCapacity	正在运行但未分配给任务的实例数	计数
UtilizedCapacity	正在运行并分配给任务的实例数	计数

使用亚马逊监控 AWS PCS 实例 CloudWatch

AWS PCS 会根据需要启动 Amazon EC2 实例，以满足您的 PCS 计算节点组中定义的扩展要求。您可以使用 Amazon 在这些实例运行时对其进行监控 CloudWatch。您可以通过登录实例并使用交互式命令行工具来检查正在运行的实例的日志。但是，默认情况下， CloudWatch 指标数据仅在实例终止后保留一段有限的时间，并且实例日志通常会与支持该实例的 EBS 卷一起删除。要保留终止后由 PCS 启动的实例的指标或日志数据，您可以使用 EC2 启动模板在实例上配置 CloudWatch 代理。本主题概述了监控正在运行的实例，并提供了如何配置永久性实例指标和日志的示例。

监控正在运行的实例

查找 AWS PCS 实例

要监控由 PCS 启动的实例，请查找与集群或计算节点组关联的正在运行的实例。然后，在给定实例的 EC2 控制台中，检查状态和警报以及监控部分。如果为这些实例配置了登录访问权限，则可以连接到这些实例并检查这些实例上的各种日志文件。有关识别哪些实例由 PCS 管理的更多信息，请参阅在中查找计算节点组实例 AWS PCS。

启用详细指标

默认情况下，每隔 5 分钟收集一次实例指标。要每隔一分钟收集指标，请在计算节点组启动模板中启用详细 CloudWatch 监控。有关更多信息，请参阅 开启详细 CloudWatch监控。

配置永久性实例指标和日志

您可以通过在实例上安装和配置 Amazon CloudWatch 代理，保留实例中的指标和日志。这包括三个主要步骤：

1. 创建 CloudWatch 代理配置。

2. 将配置存储在 PCS 实例可以检索的地方。
3. 编写一个 EC2 启动模板，用于安装 CloudWatch 代理软件、获取您的配置并使用配置启动 CloudWatch 代理。

有关更多信息，请参阅 Amazon CloudWatch 用户指南中的使用 CloudWatch 代理收集指标、日志和跟踪，以及将 Amazon EC2 启动模板与 AWS PCS。

创建 CloudWatch 代理配置

在您的实例上部署 CloudWatch 代理之前，您必须生成一个 JSON 配置文件，该文件指定要收集的指标、日志和跟踪。可以使用向导创建配置文件，也可以使用文本编辑器手动创建配置文件。将为本演示手动创建配置文件。

在安装了 AWS CLI 的计算机上，创建一个名为 config.json 的 CloudWatch 配置文件，其中包含以下内容。您也可以使用以下 URL 下载该文件的副本。

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

备注

- 示例文件中的日志路径适用于亚马逊 Linux 2。如果您的实例将使用不同的基础操作系统，请根据需要更改路径。
- 要捕获其他日志，请在下添加其他条目`collect_list`。
- 中的值`{brackets}`是模板化变量。有关支持变量的完整列表，请参阅 Amazon CloudWatch 用户指南中的手动创建或编辑 CloudWatch 代理配置文件。
- 您可以选择省略`logs`或`metrics`不想收集这些信息类型。

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

此文件指示 CloudWatch 代理监视多个文件，这些文件可能有助于诊断引导、身份验证和登录以及其他疑难解答域中的错误。这些指令包括：

- `/var/log/cloud-init.log`— 实例配置初始阶段的输出
- `/var/log/cloud-init-output.log`— 实例配置期间运行的命令的输出
- `/var/log/amazon/pcs/bootstrap.log`— 在实例配置期间运行的 PC 特定操作的输出
- `/var/log/slurmd.log`— 来自 Slurm 工作负载管理器的守护程序 slurmd 的输出
- `/var/log/messages`— 来自内核、系统服务和应用程序的系统消息
- `/var/log/secure`— 与身份验证尝试相关的日志，例如 SSH、sudo 和其他安全事件

日志文件将发送到名为的 CloudWatch 日志组`/PCSLogs/instances`。日志流是实例 ID 和日志文件基本名称的组合。该日志组的保留时间为 30 天。

此外，该文件还指示 CloudWatch 代理收集几个常用指标，按实例 ID 汇总这些指标。

存储配置

CloudWatch 代理配置文件必须存储在 PCS 计算节点实例可以访问的地方。有两种常见的方法可以做到这一点。您可以将其上传到您的计算节点组实例可以通过其实例配置文件访问的 Amazon S3 存储桶，或者，您可以将其作为 SSM 参数存储在 Amazon Systems Manager 参数存储中。

上传到 S3 存储桶

要将文件存储在 S3 中，请使用下面的 AWS CLI 命令。在运行命令之前，请进行以下替换：

- *amzn-s3-demo-bucket*替换为您自己的 S3 存储桶名称

首先，（如果您已有存储桶，则这是可选的），创建一个存储桶来存放您的配置文件。

```
aws s3 mb s3://amzn-s3-demo-bucket
```

接下来，将文件上传到存储桶。

```
aws s3 cp ./config.json s3://amzn-s3-demo-bucket/
```

作为 SSM 参数存储

要将文件存储为 SSM 参数，请使用以下命令。在运行命令之前，请进行以下替换：

- *region-code*替换为您正在使用 AWS PCS 的 AWS 区域。
- （可选）将参数*AmazonCloudWatch-PCS*替换为您自己的名称。请注意，如果您更改名称的前缀，则需要在节点组实例配置文件中专门添加对 SSM 参数的读取权限。AmazonCloudWatch-

```
aws ssm put-parameter \
   --region region-code \
   --name "AmazonCloudWatch-PCS" \
   --type String \
   --value file://config.json
```

编写 EC2 启动模板

启动模板的具体细节取决于您的配置文件存储在 S3 还是 SSM 中。

使用存储在 S3 中的配置

此脚本安装 CloudWatch 代理，从 S3 存储桶导入配置文件，然后使用该文件启动 CloudWatch 代理。用您自己的详细信息替换此脚本中的以下值：

- *amzn-s3-demo-bucket*— 您的账户可以读取的 S3 存储桶的名称
- */config.json*— 相对于存储配置的 S3 存储桶根目录的路径

```
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==--
```

节点组的 IAM 实例配置文件必须有权访问存储桶。以下是上述用户数据脚本中存储桶的 IAM 策略示例。

```
{
    "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/*"
            ]
        }
    ]
}
```

另请注意，这些实例必须允许流向 S3 和 CloudWatch终端节点的出站流量。这可以使用安全组或 VPC 终端节点来实现，具体取决于您的集群架构。

使用存储在 SSM 中的配置

此脚本安装 CloudWatch 代理，从 SSM 参数导入配置文件，然后使用该文件启动 CloudWatch 代理。用您自己的详细信息替换此脚本中的以下值：

- （可选）将参数*AmazonCloudWatch-PCS*替换为您自己的名称。

```
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==--
```

节点组的 IAM 实例策略必须CloudWatchAgentServerPolicy附加。

如果您的参数名称不是以开头，AmazonCloudWatch-则需要在节点组实例配置文件中专门添加对 SSM 参数的读取权限。以下是一个 IAM 策略示例，它说明了前缀的相关内容*DOC-EXAMPLE-PREFIX*。

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "CustomCwSsmMParamReadOnly",
      "Effect" : "Allow",
      "Action" : [
        "ssm:GetParameter"
      ],
      "Resource" : "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

另请注意，这些实例必须允许流向 SSM 和 CloudWatch终端节点的出站流量。这可以使用安全组或 VPC 终端节点来实现，具体取决于您的集群架构。

使用记录 AWS 并行计算服务API调用 AWS CloudTrail

AWS PCS与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务在中执行的操作的记录 AWS PCS。 CloudTrail 将所有API呼叫捕获 AWS PCS为事件。捕获的调用包括来自 AWS PCS 控制台的调用和对 AWS PCSAPI操作的代码调用。如果您创建跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括的事件 AWS PCS。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的事件历史记录中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向哪个请求发出 AWS PCS、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅AWS CloudTrail 用户指南。

AWS PCS信息在 CloudTrail

CloudTrail 在您创建账户 AWS 账户 时已在您的账户上启用。当活动发生在中时 AWS PCS，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在中查看、搜索和下载最近发生的事件 AWS 账户。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录您的 AWS 账户事件（包括的事件） AWS PCS，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 为以下各项配置亚马逊SNS通知 CloudTrail
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有 AWS PCS操作均由《 CloudTrail 并行计算服务参考》记录并记录在《AWS 并行计算服务API参考》中。例如，对`CreateComputeNodeGroupUpdateQueue`、和`DeleteCluster`操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用 root 还是 AWS Identity and Access Management (IAM) 用户凭据发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅CloudTrail userIdentity元素。

了解来自的 CloudTrail 日志文件条目 AWS PCS

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共API调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下示例显示了CreateQueue操作的 CloudTrail 日志条目。


```
{
    "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"
}
```

的终端节点和服务配额 AWS PCS

以下各节介绍并 AWS 行计算服务 (AWS PCS) 的终端节点和服务配额。服务配额（以前称为限制）是您的服务资源或操作的最大数量 AWS 账户。

您的每项 AWS 服务 AWS 账户 都有默认配额。除非另有说明，否则，每个限额是区域特定的。您可以请求增加某些配额，但其他一些配额无法增加。

有关更多信息，请参阅《AWS 一般参考》中的 AWS 服务限额。

目录

服务端点

区域名称	区域	端点	协议
美国东部 (弗吉尼亚北部)	us-east-1	pcs.us-east-1.amazonaws.com	HTTPS
美国东部（俄亥俄州）	us-east-2	pcs.us-east-2.amazonaws.com	HTTPS
美国西部（俄勒冈州）	us-west-2	pcs.us-west-2.amazonaws.com	HTTPS
亚太地区（新加坡）	ap-southeast-1	pcs.ap-southeast-1.amazonaws.com	HTTPS
亚太地区（悉尼）	ap-southeast-2	pcs.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	pcs.ap-northeast-1.amazonaws.com	HTTPS

区域名称	区域	端点	协议
欧洲（法兰克福）	eu-central-1	pcs.eu-central-1.amazonaws.com	HTTPS
欧洲地区（爱尔兰）	eu-west-1	pcs.eu-west-1.amazonaws.com	HTTPS
欧洲地区（斯德哥尔摩）	eu-north-1	pcs.eu-north-1.amazonaws.com	HTTPS

服务限额

名称	默认	可调整	描述
集群	5	是	每个集群的最大数量 AWS 区域。

> **Note**
>
> 默认值为由设置的初始配额 AWS。这些默认值与实际应用的限额值和最大可能的服务限额是分开的。有关更多信息，请参阅《服务限额用户指南》中的服务限额中的术语。

这些服务配额列在中的AWS 并行计算服务 (PCS) 下AWS Management Console。要为显示为可调整的值申请增加配额，请参阅 Servic e Quotas 用户指南中的申请增加配额。

> **Important**
>
> 请记得在中查看当前 AWS 区域 设置 AWS Management Console。

内部配额

以下配额为内部配额，不可调整。

名称	默认	可调整	描述
并发创建集群	1	否	每个Creating状态下的最大群集数 AWS 区域。

其他 AWS 服务的相关配额

AWS PCS使用其他 AWS 服务。这些服务的服务配额会影响您对的使用 AWS PCS。

会产生影响的亚马逊EC2服务配额 AWS PCS

- 竞价型实例请求
- 运行按需实例
- 启动模板
- 启动模板版本
- 亚马逊的EC2API请求

有关更多信息，请参阅亚马逊弹性计算云用户指南中的亚马逊EC2服务配额。

对 AWS 并行计算服务中的问题进行故障排除

以下主题为解决您在中可能遇到的一些问题提供了指导 AWS PCS。

主题

中的EC2实例在 AWS PCS重启后终止并被替换

问题概述

计算节点组中的EC2实例重启后， AWS PCS会自动终止并替换该实例。

为什么会发生这种情况

AWS PCS不支持实例重启。如果EC2实例重启，则 AWS PCS认为该实例运行状况不佳并替换它。如果 AWS PCS持续终止并替换您的实例，则可能是因为您的实例启动后有些东西会重新启动您的实例。一些示例包括通过EC2实例上的自动化重启（例如修补后自动重启）、EC2实例外部的自动化（例如网络管理应用程序）、其他 AWS 服务（例如 AWS Systems Manager）或人员手动重启。

操作

您可以查看您的`slurmctld`或`slurmd`日志，以查看您的实例是否已重启。有关更多信息，请参阅AWS PCS调度程序日志 和使用亚马逊监控 AWS PCS 实例 CloudWatch。以下示例`slurmctld`日志条目表示实例已重启：

Example

```
[2024-09-12T06:42:50.393+00:00] validate_node_specs: Node Login-1 unexpectedly rebooted
 boot_time=1726123354 last response=1726123285
```

因为正在修补而重新启动

应用补丁后，通常需要重新启动。不要将补丁直接应用于属于 AWS PCS计算节点组的EC2实例。如果您必须修补您的EC2实例，则应将补丁应用于更新后的 Amazon 系统映像 (AMI)，并更新您的计算节点组以使用更新后的节点AMI。为这些计算节点组 AWS PCS启动的新EC2实例将使用更新的（已修补的）AMI。有关更多信息，请参阅 定制 Amazon 机器映像 (AMIs) 适用于 AWS PCS。

AWS PCS 用户指南的文档历史记录

下表描述了对的文档所做的重要更改 AWS PCS。

Date	更改	文档更新	API版本已更新
2024年12月18日	已针对 Slurm 24.05 进行了更新	更新了 Slurm 24.05 支持的用户指南。有关更多信息，请参阅要为其定制AMIs构建的软件安装程序 AWS PCS 和AWS PCS示例的发行说明 AMIs。	不适用
2024年12月18日	Slurm 23.11 示例的更新NVIDIA版本 AMIs	更新了 Slurm 23.11 示例中的NVIDIA驱动程序和CUDA版本。AMIs 有关更多信息，请参阅 AWS PCS示例的发行说明 AMIs。	不适用
2024年12月17日	更新了 Slurm 安装程序	更新了 Slurm 安装程序 23.11.10-3 AMI 的主题。有关更多信息，请参阅 要为其定制AMIs构建的软件安装程序 AWS PCS。	不适用
2024 年 12 月 13 日	更新了 PCS 代理	更新了 AWS PCS代理 1.1.1-1 AMI 的主题。有关更多信息，请参阅 要为其定制AMIs构建的软件安装程序 AWS PCS。	不适用
2024 年 12 月 6 日	更新了PCS代理和 Slurm 安装程序	更新了 AWS PCS代理 1.1.0-1 和 Slurm 安装程序 23.11.10-2 AMI 的主题。有关更多信息，请参阅 要	不适用

Date	更改	文档更新	API版本已更新
		为其定制AMIs构建的软件安装程序 AWS PCS。	
2024 年 12 月 6 日	添加了有关操作系统支持的主题	有关更多信息，请参阅中支持的操作系统 AWS PCS。	不适用
2024 年 11 月 8 日	重新整理的用户指南	我们重新整理了用户指南，将主题置于顶层，将一些主题移到了自己的页面，并将相似的主题组合在一起。	不适用
2024 年 11 月 7 日	更新的AMI话题	更新了 Slurm 23.11.10 和 libjwt 17.0 AMI 的主题。有关更多信息，请参阅要为其定制AMIs构建的软件安装程序 AWS PCS 和第 3 步 — 安装 Slurm。 简化并更正了的发行说明 AMIs。有关更多信息，请参阅 AWS PCS示例的发行说明 AMIs。	不适用
2024 年 11 月 7 日	添加了有关使用加密EBS卷的新主题 AWS PCS	添加了一个主题，描述了中加密EBS卷所需的KMS密钥策略 AWS PCS。有关更多信息，请参阅 在 PCS 中使用加密 EBS 卷所必需的 KMS 密钥策略 AWS。	不适用

Date	更改	文档更新	API版本已更新
2024 年 10 月 18 日	AWS PCS特工 1.0.1-1 已发布	更新了AMI相关文档，以参考 AWS PCS代理版本 1.0.1-1。有关更多信息，请参阅要为其定制AMIs构建的软件安装程序 AWS PCS 和步骤 2-安装代 AWS PCS理。	不适用
2024 年 10 月 10 日	添加了疑难解答章节	添加了疑难解答章节，其中包含有关EC2实例在重启后自动替换的主题。有关更多信息，请参阅 对 AWS 并行计算服务中的问题进行故障排除。	不适用
2024 年 9 月 23 日	更新了使用API操作和服务管理员的最低权限	现在需要`ec2:DescribeInstanceTypeOfferings` 权限才能执行`CreateComputeNodeGroup` 和`UpdateComputeNodeGroup` API操作。有关更多信息，请参阅 AWS PCS 的最低权限。	不适用
2024 年 9 月 5 日	更新了服务管理员最低权限的示例IAM策略	有关更多信息，请参阅 服务管理员的最低权限。	不适用
2024 年 9 月 5 日	在托管策略页面JSON中添加了一个缺失的权限	这只是对文档的更正。实际的托管策略没有改变。有关更多信息，请参阅 AWSAWS 并行计算服务的托管策略。	不适用

Date	更改	文档更新	API版本已更新
2024 年 8 月 28 日	已添加托管策略页面	有关更多信息，请参阅 AWSAWS 并行计算服务的托管策略。	不适用
2024 年 8 月 28 日	AWS PCS 版本	AWS PCS用户指南的初始版本。	AWS SDK: 2024-08-28

AWS 词汇表

有关最新 AWS 术语，请参阅《AWS 词汇表 参考资料》中的AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。