使用者指南

AWS PCS

AWS PCS: 使用者指南

Table of Contents

什麼是 AWS 平行運算服務？

AWS 平行運算服務 (AWS PCS) 是一種受管服務，可讓您更輕鬆地執行和擴展高效能運算 (HPC) 工作負載，並在 AWS 上使用 Slurm 建置科學和工程模型。使用 AWS PCS 建置整合業界最佳運算、儲存、聯網和視覺化的 AWS 運算叢集。執行模擬或建置科學和工程模型。使用內建的管理和可觀測性功能，簡化叢集操作。透過讓使用者能夠在熟悉的環境中執行應用程式和任務，讓您的使用者能夠專注於研究和創新。

主題

AWS PCS 中的概念

AWS PCS 中的叢集有 1 個或多個佇列，與至少 1 個運算節點群組相關聯。任務會提交至佇列，並在運算節點群組定義的 EC2 執行個體上執行。您可以使用這些基礎來實作複雜的 HPC 架構。

叢集

叢集是管理資源和執行工作負載的資源。叢集是一種 AWS PCS 資源，可定義運算、聯網、儲存、身分和任務排程器組態的組合。您可以透過指定要使用的任務排程器（目前為 Slurm)、您想要的排程器組態、您想要管理叢集的服務控制器，以及您想要在哪些 VPC 中啟動叢集資源，來建立叢集。排程器接受並排程任務，也會啟動處理這些任務的運算節點 (EC2 執行個體）。

運算節點群組

運算節點群組是 AWS PCS 用來執行任務或提供叢集互動式存取的運算節點集合。當您定義運算節點群組時，您可以指定常見的特徵，例如 Amazon EC2 執行個體類型、執行個體數量下限和上限、目標 VPC 子網路、Amazon Machine Image (AMI)、購買選項和自訂啟動組態。 AWS PCS 使用這些設定來有效率地啟動、管理和終止運算節點群組中的運算節點。

佇列

當您想要在特定叢集上執行任務時，您可以將其提交至特定佇列（有時也稱為分割區)。任務會保留在佇列中，直到 AWS PCS 排程在運算節點群組上執行。您可以將一或多個運算節點群組與每個佇列建立關聯。需要使用 佇列，以使用任務排程器提供的各種排程政策，在基礎運算節點群組資源上排程和執行任務。使用者不會將任務直接提交至運算節點或運算節點群組。

系統管理員

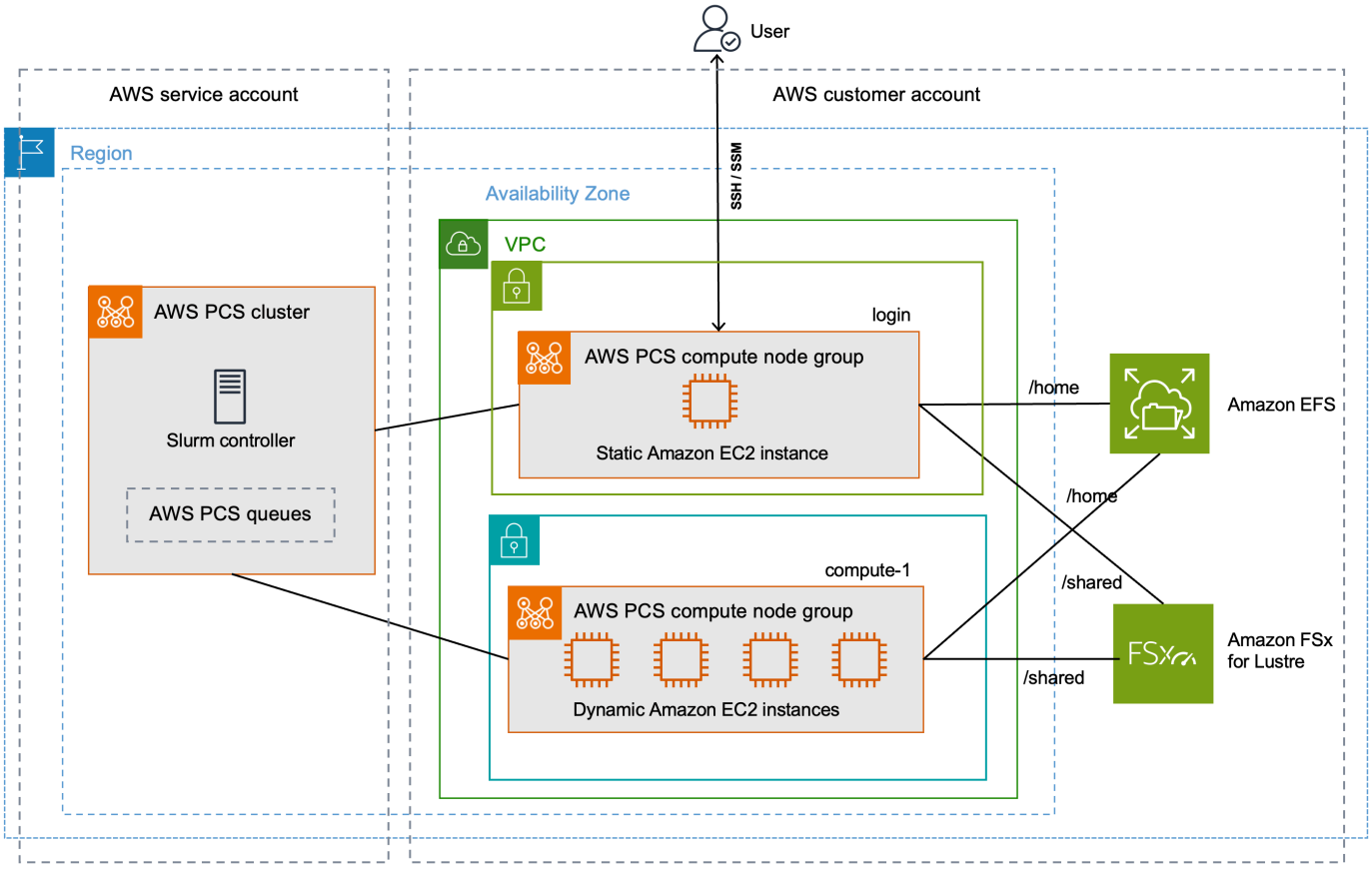
系統管理員會部署、維護和操作叢集。他們可以透過 AWS Management Console、 AWS PCS API 和 AWS SDK 存取 AWS PCS。他們可以透過 SSH 或 存取特定叢集 AWS Systems Manager，在其中執行管理任務、執行任務、管理資料，以及執行其他以 shell 為基礎的活動。如需詳細資訊，請參閱 AWS Systems Manager 文件。

最終使用者

最終使用者沒有部署或操作叢集的day-to-day責任。他們使用終端機界面 （例如 SSH) 來存取叢集資源、執行任務、管理資料，以及執行其他以 shell 為基礎的活動。

平行 AWS 運算服務入門

這是建立簡單叢集的教學課程，可用來試用 AWS PCS。下圖顯示叢集的設計。



教學課程叢集設計具有下列重要元件：

- 符合 AWS PCS 網路需求的 VPC 和子網路。
- Amazon EFS 檔案系統，將用作共用主目錄。
- Amazon FSx for Lustre 檔案系統，提供共用的高效能目錄。
- AWS PCS 叢集，提供 Slurm 控制器。
- 2 AWS PCS 運算節點群組。
  - `login` 節點群組，提供系統的 shell 型互動式存取。
  - `compute-1` 節點群組提供彈性擴展的執行個體來執行任務。
- 將任務傳送至`compute-1`節點群組中 EC2 執行個體的 1 個佇列。

叢集需要額外 AWS 的資源，例如安全群組、IAM 角色和 EC2 啟動範本，這些資源不會顯示於圖表中。

> **Note**
>
> 建議您在 Bash shell 中完成本主題中的命令列步驟。如果您不使用 Bash shell，則某些指令碼命令 (如行接續字元以及設定和使用變數的方式) 需要針對 shell 進行調整。此外，您的 Shell 的引用及轉義規則可能會有所不同。如需詳細資訊，請參閱《 第 AWS Command Line Interface 2 版使用者指南》中的引號和含有字串的常值 AWS CLI。

主題

開始使用 AWS PCS 的先決條件

請參閱下列主題，為 AWS PCS 準備您的 AWS 帳戶 和本機開發環境。

主題

- 安裝 AWS CLI 適用於 AWS PCS 的
- AWS PCS 所需的 IAM 許可

註冊 AWS 並建立管理使用者

完成下列任務以設定 AWS 平行運算服務 (AWS PCS)。

主題

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 在註冊程序完成後， 會傳送確認電子郵件給您。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center和建立 管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者 （主控台） 啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的啟用 AWS IAM Identity Center。

2. 在 IAM Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center 《 使用者指南》中的使用預設值設定使用者存取 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

  如需使用 IAM Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

安裝 AWS CLI 適用於 AWS PCS 的

您必須使用最新版本的 AWS CLI。如需詳細資訊，請參閱《 第 2 版使用者指南》中的安裝或更新至最新版本 AWS CLIAWS Command Line Interface 的 。

您必須設定 AWS CLI。如需詳細資訊，請參閱《 第 AWS Command Line Interface 2 版使用者指南》中的設定 AWS CLI 。

在命令提示字元中輸入下列命令來檢查您的 AWS CLI；它應該會顯示說明資訊。

```
aws pcs help
```

AWS PCS 所需的 IAM 許可

您使用的 IAM 安全主體必須具有使用 AWS PCS IAM 角色、服務連結角色 AWS CloudFormation、VPC 和相關資源的許可。如需詳細資訊，請參閱《 AWS Identity and Access Management 使用者指南》中的 AWS 平行運算服務的 Identity and Access Management和建立服務連結角色。您必須以同一位使用者的身分完成本指南中的所有步驟。若要檢查目前使用者，請執行以下命令：

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

使用 AWS CloudFormation 搭配 AWS PCS 教學課程

AWS PCS 教學課程有許多步驟，旨在協助您了解 AWS PCS 叢集的各個部分，以及建立叢集所需的程序。建議您至少完成教學步驟 1 次。充分了解所涉及的內容之後，您可以使用 AWS CloudFormation 來透過自動化快速建立範例叢集。

AWS CloudFormation 是一項 AWS 服務，可讓您以可預測且重複的方式建立和佈建 AWS 基礎設施部署。您可以使用 CloudFormation 範本，將範例叢集 AWS 的資源自動佈建為單一單位，稱為堆疊。您可以在完成時刪除堆疊。

如需詳細資訊，請參閱AWS CloudFormation 和 AWS PCS 入門。

建立 AWS PCS 的 VPC 和子網路

您可以使用 CloudFormation 範本建立 VPC 和子網路。使用以下 URL 下載 CloudFormation 範本，然後在AWS CloudFormation 主控台中上傳範本以建立新的 CloudFormation 堆疊。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 AWS CloudFormation 主控台。

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

在 AWS CloudFormation 主控台中開啟範本時，輸入下列選項。您可以使用範本中提供的預設值。

- 在提供堆疊名稱下：
  - 在堆疊名稱下，輸入：

```
hpc-networking
```

- 在 參數下：
  - 在 VPC 下：
    - 在 CidrBlock 下，輸入：

```
10.3.0.0/16
```

  - 在子網路 A 下：
    - 在 CidrPublicSubnetA 下，輸入：

```
10.3.0.0/20
```

    - 在 CidrPrivateSubnetA 下，輸入：

```
10.3.128.0/20
```

  - 在子網路 B 下：
    - 在 CidrPublicSubnetB 下，輸入：

```
10.3.16.0/20
```

    - 在 CidrPrivateSubnetB 下，輸入：

```
10.3.144.0/20
```

  - 在子網路 C 下：
    - 針對 ProvisionSubnetsC，選取 True
    - 在 CidrPublicSubnetC 下，輸入：

```
10.3.32.0/20
```

    - 在 CidrPrivateSubnetC 下，輸入：

```
10.3.160.0/20
```

- 在功能下：
  - 勾選核取方塊表示我確認 AWS CloudFormation 可能會建立 IAM 資源。

監控 CloudFormation 堆疊的狀態。到達 時CREATE_COMPLETE，請在新的 VPC 中找到預設安全群組的 ID。您稍後會在教學課程中使用 ID。

尋找叢集 VPC 的預設安全群組

若要在新的 VPC 中尋找預設安全群組的 ID，請遵循下列程序：

- 導覽至 Amazon VPC 主控台。
- 在 VPC 儀表板下，選取依 VPC 篩選。
  - 選擇名稱開頭為 的 VPChpc-networking。
  - 在安全下，選擇安全群組。
- 尋找名為 之群組的安全群組 IDdefault。它具有描述 `default VPC security group`。您稍後會使用 ID 來設定 EC2 啟動範本。

建立 AWS PCS 的安全群組

AWS PCS 依賴安全群組來管理進出叢集及其運算節點群組的網路流量。如需此主題的詳細資訊，請參閱 安全群組需求和考量事項。

在此步驟中，您將使用 CloudFormation 範本來建立兩個安全群組。

- 叢集安全群組，可啟用 AWS PCS 控制器、運算節點和登入節點之間的通訊。
- 傳入 SSH 安全群組，您可以選擇將其新增至登入節點以支援 SSH 存取

建立 AWS PCS 的安全群組

您可以使用 CloudFormation 範本來建立安全群組。使用以下 URL 下載 CloudFormation 範本，然後在AWS CloudFormation 主控台中上傳範本以建立新的 CloudFormation 堆疊。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 AWS CloudFormation 主控台。

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

在 AWS CloudFormation 主控台中開啟範本時，輸入下列選項。請注意，某些選項會預先填入範本中，您只需將它們保留為預設值即可。

- 在提供堆疊名稱下
  - 在堆疊名稱下，輸入：

```
getstarted-sg
```

- 在 參數下
  - 在 VpcId 下，選擇名稱開頭為 的 VPChpc-networking。
  - （選用）在 ClientIpCidr 下，為傳入 SSH 安全群組輸入更嚴格的 IP 範圍。我們建議您使用自己的 IP/子網路 (x.x.x.x/32 表示您自己的 ip，或 x.x.x.x/24 表示範圍。 將 x.x.x.x 取代為您自己的 PUBLIC IP。 您可以使用 https://ifconfig.co/ 等工具取得公有 IP)

監控 CloudFormation 堆疊的狀態。到達CREATE_COMPLETE安全群組資源時，已準備就緒。

已建立兩個安全群組，名稱為：

- cluster-getstarted-sg – 這是叢集安全群組
- inbound-ssh-getstarted-sg – 這是允許傳入 SSH 存取的安全群組

在 AWS PCS 中建立叢集

在 AWS PCS 中，叢集是用於管理資源和執行工作負載的持久性資源。您可以在新的或現有 VPC 的子網路中，為特定排程器 (AWS PCS 目前支援 Slurm) 建立叢集。叢集接受並排程任務，也會啟動處理這些任務的運算節點 (EC2 執行個體）。

若要建立叢集

1. 開啟 AWS PCS 主控台，然後選擇建立叢集。
2. 在叢集詳細資訊區段中，輸入下列欄位：
   - 叢集名稱 – 輸入 get-started
   - 排程器 – 選取 Slurm 24.11 版
   - 控制器大小 – 選取小型
3. 在聯網區段中，選取下列欄位的值：

- VPC – 選擇名為 的 VPC `hpc-networking:Large-Scale-HPC`
- 子網路 – 選取名稱開頭為 的子網路 `hpc-networking:PrivateSubnetA`
- 安全群組 – 選取名為 的叢集安全群組 `cluster-getstarted-sg`

4. 選擇 建立叢集。

> **Note**
>
> 在佈建叢集時，狀態欄位會顯示建立。建立叢集可能需要幾分鐘的時間。

在 Amazon Elastic File System 中為 AWS PCS 建立共用儲存

Amazon Elastic File System (Amazon EFS) 是一項 AWS 服務，可提供無伺服器、全彈性的檔案儲存，因此您可以共用檔案資料，而無需佈建或管理儲存容量和效能。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的什麼是 Amazon Elastic File System？。 Amazon Elastic File System

AWS PCS 示範叢集使用 EFS 檔案系統，在叢集節點之間提供共用的主目錄。在與叢集相同的 VPC 中建立 EFS 檔案系統。

建立 Amazon EFS 檔案系統

1. 前往 Amazon EFS 主控台。
2. 請確定它設定為您要嘗試 AWS PCS AWS 區域 的相同 。
3. 選擇 Create file system (建立檔案系統)。
4. 在建立檔案系統頁面上，設定下列參數：
   - 對於名稱，輸入 `getstarted-efs`。
   - 在虛擬私有雲端 (VPC) 下，選擇名為 的 VPC `hpc-networking:Large-Scale-HPC`
   - 選擇 Create (建立)。這會讓您返回檔案系統頁面。
5. 請記下檔案系統的檔案系統 ID。 `getstarted-efs`您之後將會用到此資訊。

在 Amazon FSx for Lustre 中為 AWS PCS 建立共用儲存

Amazon FSx for Lustre 可讓您輕鬆且符合成本效益地啟動和執行熱門的高效能 Lustre 檔案系統。對於速度很重要的工作負載，例如機器學習、高效能運算 (HPC)、影片處理和財務建模。如需詳細資訊，請參閱《Amazon FSx for Lustre 使用者指南》中的什麼是 Amazon FSx for Lustre？。 FSx

AWS PCS 示範叢集可以使用 FSx for Lustre 檔案系統，在叢集節點之間提供高效能的共用目錄。在與叢集相同的 VPC 中建立 FSx for Lustre 檔案系統。

建立 FSx for Lustre 檔案系統

1. 前往 Amazon FSx 主控台。
2. 請確定主控台設定為使用 AWS 區域 與您的叢集相同的 。
3. 選擇 Create file system (建立檔案系統)。
   - 對於選取檔案系統類型，選擇 Amazon FSx for Lustre，然後選擇下一步。
4. 在指定檔案系統詳細資訊頁面上，設定下列參數：
   - 檔案系統詳細資訊下
     - 對於名稱，輸入 `getstarted-fsx`。
     - 針對部署和儲存類型，選擇持久性、SSD
     - 針對每單位儲存的輸送量，選擇 125 MB/s/TiB
     - 針對儲存容量，輸入 1.2 TiB
     - 針對中繼資料組態，選擇自動
     - 針對資料壓縮類型，選擇 LZ4
   - 在網路與安全下
     - 針對虛擬私有雲端 (VPC)，選擇名為 的 VPC `hpc-networking:Large-Scale-HPC`
     - 對於 VPC 安全群組，保留名為 的安全群組 `default`
     - 針對子網路，選擇名稱開頭為 的子網路 `hpc-networking:PrivateSubnetA`
   - 將其他選項設定為其預設值。
   - 選擇 Next (下一步)。
5. 在檢閱和建立頁面上，選擇建立檔案系統。這會讓您返回檔案系統頁面。
6. 導覽至您建立之 FSx for Lustre 檔案系統的詳細資訊頁面。
7. 記下檔案系統 ID 和掛載名稱。您之後將會用到此資訊。

> **Note**
>
> 狀態欄位會顯示正在佈建檔案系統時的建立。檔案系統建立可能需要幾分鐘的時間。等到它完成，再繼續教學課程的其餘部分。

在 AWS PCS 中建立運算節點群組

運算節點群組是 AWS PCS 啟動和管理的虛擬運算節點集合 (EC2 執行個體）。當您定義運算節點群組時，您可以指定常見的特徵，例如 EC2 執行個體類型、最小和最大執行個體計數、目標 VPC 子網路、偏好的購買選項，以及自訂啟動組態。 AWS PCS 會根據這些設定，有效率地啟動、管理和終止運算節點群組中的運算節點。示範叢集使用運算節點群組來提供登入節點以供使用者存取，以及單獨的運算節點群組來處理任務。下列主題說明在叢集中設定這些運算節點群組的程序。

主題

建立 AWS PCS 的執行個體設定檔

運算節點群組在建立時需要執行個體描述檔。如果使用 AWS Management Console 為 Amazon EC2 建立角色，則主控台會自動建立執行個體描述檔，並將其命名為與角色相同的名稱。如需詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的使用執行個體設定檔。

在下列程序中，您可以使用 AWS Management Console 為 Amazon EC2 建立角色，這也會為您的運算節點群組建立執行個體描述檔。

建立角色和執行個體描述檔

- 導覽至 IAM 主控台。
- 在 Access management (存取管理) 下，請選擇 Policies (政策)。
  - 選擇 Create policy (建立政策)。
  - 在指定許可下，針對政策編輯器，選擇 JSON。
  - 將文字編輯器的內容取代為以下內容：

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

- 選擇 Next (下一步)。
- 在檢閱和建立下，針對政策名稱輸入 AWSPCS-getstarted-policy。
- 選擇 建立政策。

- 在 Access management (存取管理) 下，請選擇 Roles (角色)。
- 選擇建立角色。
- 在選取信任的實體下：
  - 針對信任的實體類型，選取AWS 服務
  - 在使用案例下，選取 EC2。
    - 然後，在選擇指定服務的使用案例下，選擇 EC2。
  - 選擇 Next (下一步)。
- 在新增許可下：
  - 在許可政策中，搜尋 AWSPCS-getstarted-policy。
  - 勾選 AWSPCS-getstarted-policy 旁的方塊，將其新增至角色。
  - 在許可政策中，搜尋 AmazonSSMManagedInstanceCore。
  - 勾選 AmazonSSMManagedInstanceCore 旁的方塊，將其新增至角色。
  - 選擇 Next (下一步)。
- 在名稱下，檢閱和建立：
  - 在角色詳細資訊下：
    - 在角色名稱中，輸入 AWSPCS-getstarted-role。
  - 選擇建立角色。

建立 AWS PCS 的啟動範本

當您建立運算節點群組時，您會提供 EC2 啟動範本，供 AWS PCS 用來設定其啟動的 EC2 執行個體。這包括設定，例如在執行個體啟動時執行的安全群組和指令碼。

在此步驟中，一個 CloudFormation 範本將用於建立兩個 EC2 啟動範本。一個範本將用於建立登入節點，另一個範本將用於建立運算節點。兩者之間的主要區別在於，登入節點可以設定為允許傳入 SSH 存取。

存取 CloudFormation 範本

使用以下 URL 下載 CloudFormation 範本，然後在AWS CloudFormation 主控台中上傳範本以建立新的 CloudFormation 堆疊。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 AWS CloudFormation 主控台。

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

使用 CloudFormation 範本建立 EC2 啟動範本

使用下列程序在 AWS CloudFormation 主控台中完成 CloudFormation 範本

- 在提供堆疊名稱下：
  - 在堆疊名稱下，輸入 `getstarted-lt`。
- 在 參數下：
  - 在安全下
    - 針對 VpcSecurityGroupId，選取叢集 VPC `default`中名為 的安全群組。
    - 針對 ClusterSecurityGroupId，選取名為 的群組 `cluster-getstarted-sg`
    - 針對 SshSecurityGroupId，選取名為 的群組 `inbound-ssh-getstarted-sg`
    - 對於 SshKeyName，選取您偏好的 SSH 金鑰對。
  - 檔案系統下
    - 對於 EfsFilesystemId，輸入您在教學中稍早建立的 EFS 檔案系統中的檔案系統 ID。
    - 對於 FSxLustreFilesystemId，輸入您在教學中稍早建立的 FSx for Lustre 檔案系統的檔案系統 ID。
    - 針對 FSxLustreFilesystemMountName，輸入相同 FSx for Lustre 檔案系統的掛載名稱。
- 選擇下一步，然後再次選擇下一步。

- 選擇提交。

監控 CloudFormation 堆疊的狀態。當它到達CREATE_COMPLETE啟動範本時，即可使用。

> **Note**
>
> 若要查看 CloudFormation 範本建立的所有資源，請開啟 AWS CloudFormation 主控台。選擇 getstarted-lt 堆疊，然後選擇 Resources (資源) 索引標籤。

在 AWS PCS 中建立登入節點的運算節點群組

運算節點群組是 AWS PCS 啟動和管理的虛擬運算節點集合 (EC2 執行個體）。當您定義運算節點群組時，您可以指定常見的特徵，例如 EC2 執行個體類型、最小和最大執行個體計數、目標 VPC 子網路、偏好的購買選項，以及自訂啟動組態。 AWS PCS 會根據這些設定，有效率地啟動、管理和終止運算節點群組中的運算節點。

在此步驟中，您將啟動靜態運算節點群組，提供叢集的互動式存取。您可以使用 SSH 或 Amazon EC2 Systems Manager (SSM) 登入，然後執行 shell 命令和管理 Slurm 任務。

建立運算節點群組

- 開啟 AWS PCS 主控台並導覽至叢集。
- 選取名為 的叢集 get-started
- 導覽至運算節點群組，然後選擇建立。
- 在運算節點群組設定區段中，提供下列項目：
  - 運算節點群組名稱 – 輸入 login。
- 在運算組態下，輸入或選取這些值：
  - EC2 啟動範本 – 選擇名稱為 的啟動範本 login-getstarted-lt
  - IAM 執行個體描述檔 – 選擇名為 的執行個體描述檔 AWSPCS-getstarted-role
  - 子網路 – 選取名稱開頭為 的子網路hpc-networking:PublicSubnetA。
  - 執行個體 – 選取 c6i.xlarge。
  - 擴展組態 – 針對最小執行個體計數，輸入 1。針對執行個體計數上限，輸入 1。
- 在其他設定下，指定下列項目：
  - AMI ID – 選取您要使用的 AMI，其名稱格式如下：

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

如需範例 AMIs的詳細資訊，請參閱 搭配 AWS PCS 使用範例 Amazon Machine Image AMIs)。

- 選擇建立運算節點群組。

狀態欄位會顯示正在佈建運算節點群組時的建立。您可以在教學課程進行時繼續下一個步驟。

建立運算節點群組以在 AWS PCS 中執行運算任務

在此步驟中，您將啟動可彈性擴展的運算節點群組，以執行提交至叢集的任務。

建立運算節點群組

- 開啟 AWS PCS 主控台並導覽至叢集。
- 選取名為 的叢集 `get-started`
- 導覽至運算節點群組，然後選擇建立。
- 在運算節點群組設定區段中，提供下列項目：
  - 運算節點群組名稱 – 輸入 `compute-1`。
- 在運算組態下，輸入或選取這些值：
  - EC2 啟動範本 – 選擇名稱為 的啟動範本 `compute-getstarted-lt`
  - IAM 執行個體描述檔 – 選擇名為 的執行個體描述檔 `AWSPCS-getstarted-role`
  - 子網路 – 選取名稱開頭為 的子網路`hpc-networking:PrivateSubnetA`。
  - 執行個體 – 選取 `c6i.xlarge`。
  - 擴展組態 – 針對最小執行個體計數，輸入 `0`。針對執行個體計數上限，輸入 `4`。
- 在其他設定下，指定下列項目：
  - AMI ID – 選取您要使用的 AMI，其名稱格式如下：

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

如需範例 AMIs的詳細資訊，請參閱 搭配 AWS PCS 使用範例 Amazon Machine Image AMIs)。

- 選擇建立運算節點群組。

狀態欄位會顯示正在佈建運算節點群組時的建立。

> **Important**
>
> 等待狀態欄位顯示作用中，然後再繼續本教學課程的下一個步驟。

建立佇列以管理 AWS PCS 中的任務

您可以將任務提交至佇列以執行任務。任務會保留在佇列中，直到 AWS PCS 排程在運算節點群組上執行。每個佇列都與一或多個運算節點群組相關聯，這些節點群組提供執行處理所需的 EC2 執行個體。

在此步驟中，您將建立佇列，使用運算節點群組來處理任務。

建立佇列

- 開啟 AWS PCS 主控台。
- 選取名為 的叢集`get-started`。
- 導覽至運算節點群組，並確認`compute-1`群組的狀態為作用中。

> **Important**
>
> `compute-1` 群組的狀態必須處於作用中狀態，才能繼續下一個步驟。

- 導覽至佇列，然後選擇建立佇列。
  - 在佇列組態區段中，提供下列值：
    - 佇列名稱 – 輸入下列項目： `demo`
    - 運算節點群組 – 選取名為 的運算節點群組`compute-1`。
- 選擇建立佇列。

在建立佇列時，狀態欄位會顯示建立。

> **Important**
>
> 等待狀態欄位顯示作用中，然後再繼續本教學課程的下一個步驟。

連線至 AWS PCS 叢集

`login` 運算節點群組的狀態變為作用中後，您就可以連線到其建立的 EC2 執行個體。

連線至登入節點

- 開啟 AWS PCS 主控台並導覽至叢集。
- 選取名為 的叢集`get-started`。
- 選擇運算節點群組。
- 導覽至名為 的運算節點群組`login`。
- 尋找運算節點群組 ID。
- 在另一個瀏覽器視窗或索引標籤中，開啟 Amazon EC2 主控台。
  - 選擇 Instances (執行個體)。
  - 使用下列標籤搜尋 EC2 執行個體。將 *`node-group-id`* 取代為上一個步驟中運算節點群組 ID 的值。應該有 1 個執行個體。

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

  - 連線至 EC2 執行個體。您可以使用 Session Manager 或 SSH。

    Session Manager

    - 選取執行個體。
    - 選擇連線。
    - 在連線至執行個體下，選取工作階段管理員。
    - 選擇連線。
    - 選擇連線。互動式終端機會在您的瀏覽器中啟動。

    SSH

    - 選取執行個體。
    - 選擇連線。
    - 在連線至執行個體下，選取 SSH 用戶端。
    - 遵循 主控台提供的指示。

      > **Note**
      >
      > 執行個體的使用者名稱**`ec2-user`**不是 `root`。

探索 AWS PCS 中的叢集環境

登入叢集後，您可以執行 shell 命令。例如，您可以變更使用者、使用共用檔案系統上的資料，以及與 Slurm 互動。

變更使用者

如果您已使用 Session Manager 登入叢集，則可能會以 身分連線`ssm-user`。這是為 Session Manager 建立的特殊使用者。使用下列命令在 Amazon Linux 2 上切換到預設使用者。如果您使用 SSH 連線，則不需要執行此操作。

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

使用共用檔案系統

您可以確認 EFS 檔案系統和 FSx for Lustre 檔案系統可與命令 搭配使用`df -h`。叢集上的輸出應類似以下內容：

```
[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 檔案系統掛載 127.0.0.1 且容量非常大。這是您在教學中稍早建立的 EFS 檔案系統。在此處寫入的任何檔案都會在叢集中所有節點`/home`的 下提供。

`/shared` 檔案系統掛載私有 IP，容量為 1.2 TB。這是您在教學中稍早建立的 FSx for Lustre 檔案系統。在此處寫入的任何檔案都將在叢集中所有節點`/shared`的 下提供。

與 Slurm 互動

主題

- 列出佇列和節點
- 顯示任務

列出佇列和節點

您可以使用 列出佇列及其相關聯的節點sinfo。叢集的輸出應類似於以下內容：

```
[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 ~]$
```

請注意名為 的分割區demo。其狀態為 ，up且最多有 4 個節點。它與節點群組中的compute-1節點相關聯。如果您編輯運算節點群組，並將執行個體數目上限增加到 8，則節點數目會讀取，8而節點清單會讀取 compute-1-[1-8]。如果您建立名為 test的第二個運算節點群組，其中包含 4 個節點，並將其新增至demo佇列，這些節點也會顯示在節點清單中。

顯示任務

您可以使用 列出系統上任何狀態的所有任務squeue。叢集的輸出應類似於以下內容：

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

當您有待定或正在執行的 Slurm 任務時，請稍後squeue再次嘗試執行。

在 AWS PCS 中執行單一節點任務

若要使用 Slurm 執行任務，請準備指定任務需求的提交指令碼，並使用 sbatch命令將其提交至佇列。通常，這是從共用目錄完成的，因此登入和運算節點具有存取檔案的共用空間。

連線至叢集的登入節點，並在其 shell 提示下執行下列命令。

- 成為預設使用者。變更為共用目錄。

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

- 使用下列命令來建立範例任務指令碼：

```
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
```

- 將任務指令碼提交至 Slurm 排程器：

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

- 提交任務時，它會以數字形式傳回任務 ID。使用該 ID 來檢查任務狀態。將下列命令中的 *job-id* 取代為從 傳回的數字sbatch。

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

Example

```
squeue --job 1
```

squeue 命令會傳回類似下列的輸出：

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

- 繼續檢查任務的狀態，直到達到 R（執行中） 狀態為止。當 squeue未傳回任何項目時，任務即完成。
- 檢查/shared目錄的內容。

```
ls -alth /shared
```

命令輸出類似於以下內容：

```
-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
```

名為 single.1.out和 的檔案是由叢集的其中一個運算節點所single.1.err撰寫。由於任務是在共用目錄 (/shared) 中執行，因此它們也可用於您的登入節點。這就是您為此叢集設定 FSx for Lustre 檔案系統的原因。

- 檢查 single.1.out 檔案的內容。

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

輸出類似以下內容：

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

在 AWS PCS 中使用 Slurm 執行多節點 MPI 任務

這些指示示範使用 Slurm 在 AWS PCS 中執行訊息傳遞界面 (MPI) 任務。

在登入節點的 shell 提示下執行下列命令。

- 成為預設使用者。變更為其主目錄。

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

- 使用 C 程式設計語言建立原始程式碼。

```
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
```

- 載入 OpenMPI 模組。

```
module load openmpi
```

- 編譯 C 程式。

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

- 撰寫 Slurm 任務提交指令碼。

```
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
```

- 變更為共用目錄。

```
cd /shared
```

- 提交任務指令碼。

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

- 使用 squeue監控任務，直到任務完成為止。
- 檢查 的內容multi.out：

```
cat multi.out
```

輸出類似如下。請注意，每個排名都有自己的 IP 地址，因為它在不同的節點上執行。

```
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
```

刪除 AWS PCS AWS 的資源

完成您為本教學課程建立的叢集和節點群組後，您應該刪除您建立的資源。

> **Important**
>
> 您可以針對 中執行的所有資源收取帳單費用 AWS 帳戶

刪除您為此教學課程建立的 AWS PCS 資源

- 開啟 AWS PCS 主控台。
- 導覽至名為 get-started 的叢集。
- 導覽至佇列區段。
- 選取名為示範的佇列。
- 選擇 刪除 。

> **Important**
>
> 等待佇列刪除後再繼續。

- 導覽至運算節點群組區段。
- 選取名為 compute-1 的運算節點群組。
- 選擇 刪除 。
- 選取名為登入的運算節點群組。
- 選擇 刪除 。

> **Important**
>
> 請等到兩個運算節點群組都已刪除，再繼續。

- 在入門的叢集詳細資訊頁面中，選擇刪除。

> **Important**
>
> 等待叢集刪除後，再繼續後續步驟。

刪除您為此教學課程建立的其他 AWS 資源

- 開啟 IAM 主控台。
  - 選擇角色。
  - 選取名為 AWSPCS-getstarted-role 的角色，然後選擇刪除。
  - 刪除角色之後，請選擇政策。
  - 選取名為 AWSPCS-getstarted-policy 的政策，然後選擇刪除。
- 開啟 AWS CloudFormation 主控台。
  - 選取名為 getstarted-lt 的堆疊。
  - 選擇 刪除 。

> **Important**
> 等待堆疊刪除再繼續。

- 開啟 Amazon EFS 主控台。
  - 選擇檔案系統。
  - 選取名為 getstarted-efs 的檔案系統。
  - 選擇 刪除 。

> **Important**
> 等待檔案系統刪除，然後再繼續。

- 開啟 Amazon FSx 主控台。
  - 選擇檔案系統。
  - 選取名為 getstarted-fsx 的檔案系統。
  - 選擇 刪除 。

> **Important**
> 等待檔案系統刪除，然後再繼續。

- 開啟 AWS CloudFormation 主控台。
  - 選取名為 getstarted-sg 的堆疊。

- 選擇 刪除 。
- 開啟 AWS CloudFormation 主控台。
  - 選取名為 hpc-networking 的堆疊。
  - 選擇 Delete (刪除)。

AWS CloudFormation 和 AWS PCS 入門

您可以使用 AWS CloudFormation 來建立 AWS PCS 叢集。 AWS CloudFormation 可讓您以可預測且重複的方式建立和佈建 AWS 基礎設施部署。您可以使用 從許多 AWS 服務 AWS CloudFormation 自動佈建資源，在 中建置高度可靠、可擴展且符合成本效益的應用程式， AWS 雲端 而無需建立和設定基礎 AWS 基礎設施。 AWS CloudFormation 可讓您使用範本檔案，以單一單位形式建立和刪除資源集合，稱為堆疊。如需詳細資訊 AWS CloudFormation，請參閱AWS CloudFormation 《 使用者指南》中的什麼是 AWS CloudFormation？。如需 中 AWS PCS 資源類型的詳細資訊 AWS CloudFormation，請參閱AWS CloudFormation 《 使用者指南》中的 AWS PCS 資源類型參考。

主題

- 使用 AWS CloudFormation 建立範例 AWS PCS 叢集
- 連線至使用 建立的 AWS PCS 叢集 AWS CloudFormation
- 在 中清除 AWS PCS 叢集 AWS CloudFormation
- AWS PCS CloudFormation 範本的一部分
- AWS CloudFormation 建立範例 AWS PCS 叢集的 範本

使用 AWS CloudFormation 建立範例 AWS PCS 叢集

下列程序使用 中的 CloudFormation 範本 AWS Management Console 來建立範例 AWS PCS 叢集。如需 的詳細資訊 AWS CloudFormation，請參閱AWS CloudFormation 《 使用者指南》中的什麼是 AWS CloudFormation？。如需 中 AWS PCS 資源類型的詳細資訊 AWS CloudFormation，請參閱AWS CloudFormation 《 使用者指南》中的 AWS PCS 資源類型參考。

建立範例叢集

1. 選擇要在 中 AWS 區域 建立叢集的 （ 連結會使用 範本開啟 CloudFormation 主控台 ）：
   - US East (N. Virginia) (美國東部 (維吉尼亞北部)) (us-east-1)
   - US East (Ohio) (美國東部 (俄亥俄)) (us-east-2)
   - US West (Oregon) (美國西部 (奧勒岡)) (us-west-2)
   - Asia Pacific (Singapore) (亞太區域 (新加坡)) (ap-southeast-1)
   - Asia Pacific (Sydney) (亞太區域 (雪梨)) (ap-southeast-2)
   - Asia Pacific (Tokyo) (亞太區域 (東京)) (ap-northeast-1)

- 歐洲（法蘭克福）(eu-central-1)
- 歐洲（愛爾蘭）(eu-west-1)
- 歐洲（倫敦）(eu-west-2)
- 歐洲（斯德哥爾摩）(eu-north-1)
- AWS GovCloud（美國東部）(us-gov-east-1)
- AWS GovCloud（美國西部）(us-gov-west-1)

2. 在提供堆疊名稱下，輸入描述性名稱。這是 CloudFormation 堆疊的名稱。範本使用此值做為 AWS PCS 叢集的名稱。
3. 在參數下：
   a. 在 SlurmVersion 下，選擇您希望叢集使用的 Slurm 版本。
   b. 在 NodeArchitecture 下，選擇 x86 以部署使用 x86_64 相容執行個體的叢集，或選擇 Graviton 以使用 Arm64 執行個體。
   c. 針對 KeyName，選擇 SSH 金鑰對以存取叢集登入節點。請確定您擁有所選金鑰對的 PEM 檔案。
   d. 對於 ClientIpCidr，以 CIDR 格式輸入 IP 範圍，以控制對登入節點的存取。

      > **Warning**
      > 的預設值`0.0.0.0/0`允許從所有 IP 地址存取 。

   e. 將 HpcRecipesS3Bucket 和 HpcRecipesBranch 的值保留為預設值。
4. 在功能和轉換下：
   a. 選取核取方塊以確認 AWS CloudFormation 將建立 IAM 資源。
   b. 選取核取方塊以確認 AWS CloudFormation 將使用自訂名稱建立 IAM 資源。
   c. 選取核取方塊以確認`CAPABILITY_AUTO_EXPAND`新堆疊。如需詳細資訊，請參閱 AWS CloudFormation API 參考中的 CreateStack。
5. 選擇建立堆疊。
6. 監控堆疊的狀態。您可以在堆疊的狀態為 之後連線到叢集`CREATE_COMPLETE`。

連線至使用 建立的 AWS PCS 叢集 AWS CloudFormation

從 AWS CloudFormation 範本建立 AWS PCS 叢集之後，您可以使用 AWS PCS 主控台 （ 在 中 AWS Management Console) 來管理叢集。您也可以連線至 叢集的 1 個登入節點，以管理叢集、執行任務和管理資料。 AWS CloudFormation 堆疊提供連結，可讓您用來連線至叢集。

連線至您的叢集

1. 開啟 AWS CloudFormation 主控台
2. 選擇您建立的堆疊。
3. 選擇堆疊的輸出索引標籤。

   堆疊提供下列連結：

   - PcsConsoleUrl — 選擇此連結以開啟已選取叢集的 AWS PCS 主控台。您可以使用它來探索叢集、節點群組和佇列組態。
   - Ec2ConsoleUrl — 選擇此連結以開啟 Amazon EC2 主控台，篩選以顯示叢集的登入節點群組管理的執行個體。

     在此檢視中，您可以選取執行個體，然後選擇連線。範例叢集的執行個體支援 Web 瀏覽器中的傳入 SSH 和 AWS Systems Manager 連線。如需詳細資訊，請參閱連線至 AWS PCS 叢集。

     連線至登入執行個體後，您可以遵循 的教學課程探索 AWS PCS 中的叢集環境。

在 中清除 AWS PCS 叢集 AWS CloudFormation

如果您使用 AWS CloudFormation 來建立 AWS PCS 叢集，您可以開啟AWS CloudFormation 主控台並刪除堆疊，以刪除叢集及其所有相關聯的資源。

> **Important**
>
> 對於範例叢集，如果您在叢集中建立其他運算節點群組或佇列 （ 除了範本 CloudFormation 範本建立的 `login`和 `compute-1`群組之外），您必須先使用 AWS PCS 主控台或刪除這些資源 AWS CLI ，再刪除 CloudFormation 堆疊。如需詳細資訊，請參閱在 AWS PCS 中刪除叢集。

AWS PCS CloudFormation 範本的一部分

CloudFormation 範本有 1 個或多個區段，每個區段都用於特定用途。 會在範本中 AWS CloudFormation 定義標準格式、語法和語言。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 CloudFormation 範本。

CloudFormation 範本可高度自訂，因此其格式可能會有所不同。若要了解 CloudFormation 範本建立 AWS PCS 叢集的必要部分，建議您檢查我們提供的範例範本，以建立範例叢集。本主題簡短說明該範例範本的章節。

> **Important**
>
> 本主題中的程式碼範例尚未完成。出現省略符號 (`[...]`) 表示未顯示其他程式碼。若要下載完整的 YAML 格式 CloudFormation 範本，請參閱 AWS CloudFormation 建立範例 AWS PCS 叢集的 範本。

內容

標頭

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

`AWSTemplateFormatVersion` 會識別範本符合的範本格式版本。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的 CloudFormation 範本格式版本語法。

`Transform` 指定 CloudFormation 用來處理範本的巨集。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的 CloudFormation 範本轉換一節。`AWS::Serverless-2016-10-31` 轉換可讓 AWS CloudFormation 處理以 AWS Serverless

Application Model (AWS SAM) 語法撰寫的範本。如需詳細資訊，請參閱AWS CloudFormation 《使用者指南》中的AWS::Serverless轉換。

中繼資料

```
### Stack metadata
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: PCS Cluster configuration
        Parameters:
          - SlurmVersion
          - ManagedAccounting
          - AccountingPolicyEnforcement
      - Label:
          default: PCS ComputeNodeGroups configuration
        Parameters:
          - NodeArchitecture
          - KeyName
          - ClientIpCidr
      - Label:
          default: HPC Recipes configuration
        Parameters:
          - HpcRecipesS3Bucket
          - HpcRecipesBranch
```

CloudFormation 範本的 metadata區段提供範本本身的相關資訊。範例範本會建立使用 AWS PCS 的完整高效能運算 (HPC) 叢集。範例範本的中繼資料區段會宣告參數，以控制 如何 AWS CloudFormation 啟動 （ 佈建 ） 對應的堆疊。有些參數可控制架構選擇 (NodeArchitecture)、Slurm 版本 (SlurmVersion) 和存取控制 (KeyName 和 ClientIpCidr)。

參數

Parameters 本節定義範本的自訂參數。 AWS CloudFormation 使用這些參數定義來建構和驗證您從此範本啟動堆疊時與 互動的格式。

```
Parameters:

  NodeArchitecture:
    Type: String
    Default: x86
```

```
    AllowedValues:
      - x86
      - Graviton
    Description: Processor architecture for the login and compute node instances

  SlurmVersion:
    Type: String
    Default: 24.11
    Description: Version of Slurm to use
    AllowedValues:
         - 24.05
         - 24.11

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

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

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

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

```
    ConstraintDescription: Value must be a valid IP or network range of the form
 x.x.x.x/x.

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

映射項目

Mappings 區段定義金鑰/值對，根據特定條件或相依性指定值。

```
Mappings:

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

資源

Resources 區段會宣告要佈建和設定作為堆疊一部分 AWS 的資源。

```
Resources:

  [...]
```

範本會以 layer 形式佈建範例叢集基礎設施。它從 VPC 組態Networking的 開始。儲存由雙系統提供： EfsStorage 用於共用儲存， FSxLStorage用於高效能儲存。核心叢集是透過 建立PCSCluster。

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

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

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

  [...]

  # Cluster
  PCSCluster:
    Type: AWS::PCS::Cluster
    Properties:
      Name: !Sub '${AWS::StackName}'
      Size: SMALL
      Scheduler:
        Type: SLURM
```

```
        Version: !Ref SlurmVersion
      Networking:
        SubnetIds:
          - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        SecurityGroupIds:
          - !GetAtt [ PCSSecurityGroup, Outputs.ClusterSecurityGroupId ]
```

對於運算資源，範本會建立兩個節點群組：PCSNodeGroupLogin適用於單一登入節點，以及PCSNodeGroupCompute適用於最多四個運算節點。許可PCSInstanceProfile和PCSLaunchTemplate執行個體組態支援這些節點群組。

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

  PCSLaunchTemplate:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        VpcDefaultSecurityGroupId: !GetAtt [ Networking, Outputs.SecurityGroup ]
        ClusterSecurityGroupId: !GetAtt [ PCSSecurityGroup,
 Outputs.ClusterSecurityGroupId ]
        SshSecurityGroupId: !GetAtt [ PCSSecurityGroup,
 Outputs.InboundSshSecurityGroupId ]
        EfsFilesystemSecurityGroupId: !GetAtt [ EfsStorage, Outputs.SecurityGroupId ]
        FSxLustreFilesystemSecurityGroupId: !GetAtt [ FSxLStorage,
 Outputs.FSxLustreSecurityGroupId ]
        SshKeyName: !Ref KeyName
        EfsFilesystemId: !GetAtt [ EfsStorage, Outputs.EFSFilesystemId ]
        FSxLustreFilesystemId: !GetAtt [ FSxLStorage, Outputs.FSxLustreFilesystemId ]
        FSxLustreFilesystemMountName: !GetAtt [ FSxLStorage,
 Outputs.FSxLustreMountName ]
```

```
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/cfn-pcs-lt-efs-fsxl.yaml'

  # Compute Node groups - Login Nodes
  PCSNodeGroupLogin:
    Type: AWS::PCS::ComputeNodeGroup
    Properties:
      ClusterId: !GetAtt [PCSCluster, Id]
      Name: login
      ScalingConfiguration:
        MinInstanceCount: 1
        MaxInstanceCount: 1
      IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
      CustomLaunchTemplate:
        TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.LoginLaunchTemplateId ]
        Version: 1
      SubnetIds:
        - !GetAtt [ Networking, Outputs.DefaultPublicSubnet ]
      AmiId: !GetAtt [PcsSampleAmi, AmiId]
      InstanceConfigs:
        - InstanceType: !FindInMap [ Architecture, LoginNodeInstances, !Ref
 NodeArchitecture ]

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

任務排程是透過 處理PCSQueueCompute。

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

AMI 選擇會透過 PcsAMILookupFn Lambda 函數和相關資源自動執行。

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

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

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

輸出

範本會透過 ClusterId、 和 輸出叢集識別和管理 URLsPcsConsoleUrlEc2ConsoleUrl。

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

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

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

AWS CloudFormation 建立範例 AWS PCS 叢集的 範本

AWS 區域 名稱	AWS 區域	檢視來源	在 中檢視 AWS Infrastructure Composer	啟動堆疊
美國東部 (維吉尼亞北部)	us-east-1	下載 YAML	在 中檢視 AWS Infrastructure Composer	

AWS 區域 名稱	AWS 區域	檢視來源	在 中檢視 AWS Infrastructure Composer	啟動堆疊
美國東部 (俄亥俄)	us-east-2	下載 YAML	在 中檢視 AWS Infrastructure Composer	
美國西部 (奧勒岡)	us-west-2	下載 YAML	在 中檢視 AWS Infrastructure Composer	
亞太區域 (新加坡)	ap-southeast-1	下載 YAML	在 中檢視 AWS Infrastructure Composer	
亞太區域 (雪梨)	ap-southeast-2	下載 YAML	在 中檢視 AWS Infrastructure Composer	
亞太區域 (東京)	ap-northeast-1	下載 YAML	在 中檢視 AWS Infrastructure Composer	
歐洲 (法蘭克福)	eu-central-1	下載 YAML	在 中檢視 AWS Infrastructure Composer	
歐洲 (愛爾蘭)	eu-west-1	下載 YAML	在 中檢視 AWS Infrastructure Composer	
歐洲 (倫敦)	eu-west-2	下載 YAML	在 中檢視 AWS Infrastructure Composer	
Europe (Stockholm)	eu-north-1	下載 YAML	在 中檢視 AWS Infrastructure Composer	

AWS 區域 名稱	AWS 區域	檢視來源	在 中檢視 AWS Infrastructure Composer	啟動堆疊
AWS GovCloud（美國東部）	us-gov-east-1	下載 YAML	不支援	
AWS GovCloud（美國西部）	us-gov-west-1	下載 YAML	不支援	

AWS PCS 叢集

AWS PCS 叢集包含下列元件：

- HPC 系統排程器軟體的受管執行個體，例如 Slurm 控制常駐程式 (`slurmctld`)。
- 與 HPC 系統排程器整合以佈建和管理 Amazon EC2 執行個體的元件。
- 與 HPC 系統排程器整合的元件，可將日誌和指標傳輸至 Amazon CloudWatch。

這些元件會在 管理的帳戶中執行 AWS。它們共同管理您客戶帳戶中的 Amazon EC2 執行個體。 AWS PCS 會在您的 Amazon VPC 子網路中佈建彈性網路介面，以提供從排程器軟體到 Amazon EC2 執行個體的連線 （ 例如，支援在其上排程批次任務，並允許使用者執行排程器命令來列出和管理這些任務 ）。

主題

在 AWS 平行運算服務中建立叢集

本主題提供可用選項的概觀，並說明您在 AWS 平行運算服務 (AWS PCS) 中建立叢集時應考量的事項。如果這是您第一次建立 AWS PCS 叢集，建議您遵循 平行 AWS 運算服務入門。本教學課程可協助您建立運作中的 HPC 系統，而無需擴展至所有可用的選項和系統架構。

先決條件

- 符合AWS PCS 網路要求的現有 VPC 和子網路。部署叢集以供生產使用之前，建議您先徹底了解 VPC 和子網要求。若要建立 VPC 和子網路，請參閱 為您的 AWS PCS 叢集建立 VPC。
- 具有建立和管理 AWS PCS 資源許可的 IAM 主體。如需詳細資訊，請參閱AWS 平行運算服務的 Identity and Access Management。

建立 AWS PCS 叢集

您可以使用 AWS Management Console 或 AWS CLI 來建立叢集。

AWS Management Console

建立叢集

1. 在 https：//https://console.aws.amazon.com/pcs/home#/clusters 開啟 AWS PCS 主控台，然後選擇建立叢集。
2. 在叢集設定區段中，輸入下列欄位：
   - 叢集名稱 – 叢集的名稱。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 40 個字元。名稱在 中必須是唯一的 AWS 帳戶 ， AWS 區域 且您要在其中建立叢集。
   - 排程器 – 選擇排程器和版本。如需詳細資訊，請參閱AWS PCS 中的 Slurm 版本。
   - 控制器大小 – 選擇控制器的大小。這決定 AWS PCS 叢集可以管理多少並行任務和運算節點。您只能在建立叢集時設定控制器大小。如需調整大小的詳細資訊，請參閱AWS PCS 中的叢集大小。
3. 在聯網區段中，選取下列欄位的值：
   - VPC – 選擇符合 AWS PCS 要求的現有 VPC。如需詳細資訊，請參閱AWS PCS VPC 和子網路需求和考量事項。建立叢集之後，您無法變更其 VPC。如果未列出VPCs，您必須先建立一個。
   - 子網路 – 列出所選 VPC 中的所有可用子網路。選擇符合 AWS PCS 子網路需求的子網路。如需詳細資訊，請參閱AWS PCS VPC 和子網路需求和考量事項。我們建議您選取私有子網路，以避免您的排程器端點暴露至公有網際網路。
   - 安全群組 – 指定您希望 AWS PCS 與其為叢集建立的網路介面建立關聯的安全群組 （多個）。您必須選取至少一個安全群組，允許叢集及其運算節點之間的通訊。您可以選取快速建立安全群組，讓 AWS PCS 在您選取的 VPC 中建立具有必要組態的安全群組，或選取現有的安全群組。如需詳細資訊，請參閱安全群組需求和考量事項。
4. （選用） 在 Slurm 會計組態區段中，您可以啟用 Slurm 會計並設定會計參數。如需詳細資訊，請參閱AWS PCS 中的低語會計。
5. （選用） 在 Slurm 組態區段中，您可以指定覆寫 AWS PCS 所設定預設值的 Slurm 組態選項：
   - 縮減閒置時間 – 這可控制動態佈建的運算節點在任務完成或終止後保持作用中狀態的時間。將此值設為較長值可能會讓後續任務更可能在節點上執行，但可能導致成本增加。較短的值將降低成本，但可能會增加 HPC 系統佈建節點的時間比例，而不是在其上執行任務。

- Prolog – 這是運算節點群組執行個體上 prolog 指令碼目錄的完整路徑。這對應至 Slurm 中的 Prolog 設定。請注意，這必須是目錄，而不是特定可執行檔的路徑。
- Epilog – 這是運算節點群組執行個體上 epilog 指令碼目錄的完整路徑。這對應至 Slurm 中的 Epilog 設定。請注意，這必須是目錄，而不是特定可執行檔的路徑。
- 選取類型參數 – 這有助於控制 Slurm 使用的資源選取演算法。將此值設為 `CR_CPU_Memory`會啟用記憶體感知排程，而設為 `CR_CPU`則會啟用僅限 CPU 排程。此參數對應至 Slurm 中的 SelectTypeParameters 設定，其中 `select/cons_tres` 由 AWS PCS `SelectType` 設定為 。

6. （選用）在標籤下，將任何標籤新增至 AWS PCS 叢集。
7. 選擇 建立叢集。當 AWS PCS 建立叢集`Creating`時，狀態欄位會顯示 。此程序需要幾分鐘的時間。

> **Important**
>
> 每個 AWS 區域 只能有一個`Creating`處於 狀態的叢集 AWS 帳戶。 AWS 如果您嘗試建立叢集時已有叢集處於 `Creating` 狀態，PCS 會傳回錯誤。

AWS CLI

建立叢集

1. 使用下列命令建立您的叢集。執行命令之前，請執行下列替換：
   - 將*##*取代為您要 AWS 區域 在其中建立叢集的 ID，例如 `us-east-1`。
   - 使用叢集的名稱取代 *`my-cluster`*。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 40 個字元。名稱在建立叢集的 AWS 區域 和 AWS 帳戶 中必須是唯一的。
   - 將 *`24.11`* 取代為任何支援的 Slurm 版本。

   > **Note**
   >
   > AWS PCS 目前支援 Slurm 24.11 和 24.05。

- 將 *SMALL* 取代為任何支援的叢集大小。這決定 AWS PCS 叢集可以管理多少並行任務和運算節點。只有在建立叢集時才能設定。如需調整大小的詳細資訊，請參閱AWS PCS 中的叢集大小。
- 將 的值取代`subnetIds`為您自己的值。我們建議您選取私有子網路，以避免您的排程器端點暴露至公有網際網路。
- 指定`securityGroupIds`您希望 AWS PCS 與其為叢集建立的網路介面建立關聯的 。安全群組必須與叢集位於相同的 VPC 中。您必須選取至少一個安全群組，允許叢集及其運算節點之間的通訊。如需詳細資訊，請參閱安全群組需求和考量事項。
- 或者，您可以提供自訂 KMS 金鑰，以使用 加密控制器的資料`--kms-key-id` *kms-key*。*kms-key* 以現有的 KMS ARN、金鑰 ID 或別名取代 。請注意，用於建立叢集的帳戶必須具有自訂 KMS 金鑰`kms:Decrypt`的權限。

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=24.11 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

- 或者，您可以新增 `--slurm-configration`選項來自訂 Slurm 行為，並指定 Slurm 組態選項。下列範例會將縮減閒置時間設定為 60 分鐘 (3600 秒），啟用 Slurm 會計，並將`slurm.conf`設定指定為 的值`slurmCustomSettings`。如需詳細資訊，請參閱AWS PCS 中的低語會計。

> **Note**
>
> Slurm 24.11 或更新版本支援會計。

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=24.11 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
    --slurm-configuration
 scaleDownIdleTimeInSeconds=3600,accounting='{mode=STANDARD}',slurmCustomSettings='[{p
```

2. 佈建叢集可能需要幾分鐘的時間。您可以使用下列命令來查詢叢集的狀態。在叢集的狀態欄位為 之前，請勿繼續建立佇列或運算節點群組ACTIVE。

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

> **Important**
>
> 每個 AWS 區域 只能有一個Creating處於 狀態的叢集 AWS 帳戶。 AWS 如果您嘗試建立叢集時已有叢集處於 Creating 狀態，PCS 會傳回錯誤。

您叢集的建議後續步驟

- 新增運算節點群組。
- 新增佇列。
- 啟用記錄。

在 AWS PCS 中刪除叢集

本主題提供如何刪除 AWS PCS 叢集的概觀。

刪除 AWS PCS 叢集時的考量事項

- 與叢集相關聯的所有佇列都必須刪除，才能刪除叢集。如需詳細資訊，請參閱在 AWS PCS 中刪除佇列。
- 所有與叢集相關聯的運算節點群組都必須刪除，才能刪除叢集。如需詳細資訊，請參閱在 AWS PCS 中刪除運算節點群組。

刪除叢集

您可以使用 AWS Management Console 或 AWS CLI 來刪除叢集。

AWS Management Console

刪除叢集

1. 開啟 AWS PCS 主控台。
2. 選取要刪除的叢集。
3. 選擇 刪除 。
4. 叢集狀態欄位會顯示 Deleting。此需要幾分鐘的時間來完成。

AWS CLI

刪除叢集

1. 使用下列命令來刪除叢集，並包含這些取代：
   - AWS 區域 將 *region-code* 取代為您的叢集所在的 。
   - 以叢集的名稱或 ID 取代 *my-cluster*。

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

2. 刪除叢集可能需要幾分鐘的時間。您可以使用下列命令來檢查叢集的狀態。

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

AWS PCS 中的叢集大小

AWS PCS 提供高可用性且安全的叢集，同時自動化修補、節點佈建和更新等關鍵任務。

當您建立叢集時，您可以根據兩個因素來選取其大小：

- 其將管理的運算節點數量
- 您預期在叢集上執行的作用中和佇列任務數量

> **Important**
>
> 您無法在建立叢集之後變更叢集大小。如果您需要變更大小，則必須建立新的叢集。

Slurm 叢集大小	受管執行個體數量	作用中和佇列任務的數量
小型	最多 32 個	最多 256 個
中	最多 512 個	高達 8192
大型	截至 2048 年	高達 16384

範例

- 如果您的叢集最多有 24 個受管執行個體，且最多執行 100 個任務，請選擇小型。
- 如果您的叢集最多有 24 個受管執行個體，且最多執行 1000 個任務，請選擇中。
- 如果您的叢集最多有 1000 個受管執行個體，且最多執行 100 個任務，請選擇大型。
- 如果您的叢集最多有 1000 個受管執行個體，且最多執行 10，000 個任務，請選擇大型。

在 AWS PCS 中使用叢集秘密

在建立叢集時， AWS PCS 會建立叢集秘密，這是連線到叢集上的任務排程器所需的秘密。您也會建立 AWS PCS 運算節點群組，其會定義要啟動的執行個體集，以回應擴展事件。 AWS PCS 會使用叢集秘密來設定這些運算節點群組啟動的執行個體，讓他們可以連線至任務排程器。在某些情況下，您可能想要手動設定 Slurm 用戶端。範例包括建置持久性登入節點，或設定具有任務管理功能的工作流程管理員。

AWS PCS 會將叢集秘密儲存為受管秘密，並在 `pcs!` 中使用字首 AWS Secrets Manager。秘密的成本包含在使用 AWS PCS 的費用中。

> **Warning**
>
> 請勿修改叢集秘密。如果您修改叢集秘密， AWS PCS 將無法與叢集通訊。 AWS PCS 不支援叢集秘密的輪換。如果您需要修改叢集秘密，則必須建立新的叢集。

內容

- 使用 AWS Secrets Manager 尋找叢集秘密
- 使用 AWS PCS 尋找叢集秘密
- 取得 Slurm 叢集秘密

使用 AWS Secrets Manager 尋找叢集秘密

AWS Management Console

1. 導覽至 Secrets Manager 主控台。
2. 選擇秘密，然後搜尋pcs!字首。

> **Note**
>
> AWS PCS 叢集秘密的名稱格式為 ，pcs!slurm-secret-*cluster-id*其中 *cluster-id*是 AWS PCS 叢集 ID。

AWS CLI

每個 AWS PCS 叢集秘密也會以 標記aws:pcs:*cluster-id*。您可以使用以下命令取得叢集的秘密 ID。在執行 命令之前進行這些替換：

- *region* 將 取代 AWS 區域 為 以在 中建立叢集，例如 us-east-1。
- *cluster-id* 將 取代為 AWS PCS 叢集的 ID，以尋找其叢集秘密。

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

使用 AWS PCS 尋找叢集秘密

您可以使用 AWS CLI 尋找 AWS PCS 叢集秘密的 ARN。輸入以下命令，進行下列取代：

- *region* 將 取代 AWS 區域 為 以在 中建立叢集，例如 us-east-1。
- *my-cluster* 以叢集的名稱或識別符取代 。

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

下列範例輸出來自 get-cluster命令。您可以使用 secretArn和 secretVersion一起取得秘密。

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

取得 Slurm 叢集秘密

您可以使用 Secrets Manager 取得 Slurm 叢集秘密的目前 base64 編碼版本。下列範例使用 AWS CLI。在執行 命令之前，請先進行下列取代。

- *region* 將 取代 AWS 區域 為 以在 中建立叢集，例如 us-east-1。
- secretArn 從 AWS PCS 叢集*secret-arn*將 取代為 。

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

如需有關如何使用 Slurm 叢集秘密的資訊，請參閱 使用獨立執行個體做為 AWS PCS 登入節點。

許可

您可以使用 IAM 主體來取得 Slurm 叢集秘密。IAM 主體必須具有讀取秘密的許可。如需詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的角色術語和概念。

下列範例 IAM 政策允許存取範例叢集秘密。

```
{
    "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 運算節點群組

AWS PCS 運算節點群組是節點 (Amazon EC2 執行個體） 的邏輯集合。這些節點可用於執行運算任務，以及提供 HPC 系統的互動式、殼層型存取。運算節點群組包含建立節點的規則，包括要使用的 Amazon EC2 執行個體類型、要執行的執行個體數量、是否使用 Spot 執行個體或隨需執行個體、要使用的子網路和安全群組，以及如何在啟動時設定每個執行個體。當這些規則更新時， AWS PCS 會更新與運算節點群組相關聯的資源以符合。

主題

在 AWS PCS 中建立運算節點群組

本主題提供可用選項的概觀，並說明在 AWS 平行運算服務 (AWS PCS) 中建立運算節點群組時應考量的事項。如果這是您第一次在 AWS PCS 中建立運算節點群組，我們建議您遵循 中的教學課程平行 AWS 運算服務入門。本教學課程可協助您建立運作中的 HPC 系統，而無需擴展到所有可用的選項和系統架構。

先決條件

- 有足夠的服務配額，可在您的 中啟動所需數量的 EC2 執行個體 AWS 區域。您可以使用 AWS Management Console來檢查和請求提高您的服務配額。
- 符合 AWS PCS 聯網需求的現有 VPC 和子網路 (s)。我們建議您在部署叢集以供生產使用之前，先徹底了解這些需求。如需詳細資訊，請參閱AWS PCS VPC 和子網路需求和考量事項。您也可以使用 CloudFormation 範本建立 VPC 和子網路。 為 CloudFormation 範本 AWS 提供 HPC 配方。如需詳細資訊，請參閱 GitHub 上的 aws-hpc-recipes。
- IAM 執行個體描述檔，具有呼叫 AWS PCS `RegisterComputeNodeGroupInstance` API 動作和存取節點群組執行個體所需任何其他 AWS 資源的許可。如需詳細資訊，請參閱平行運算服務的 IAM AWS 執行個體描述檔。
- 節點群組執行個體的啟動範本。如需詳細資訊，請參閱搭配 AWS PCS 使用 Amazon EC2 啟動範本。

- 若要建立使用 Amazon EC2 Spot 執行個體的運算節點群組，您必須在您的 中具有 AWSServiceRoleForEC2Spot 服務連結角色 AWS 帳戶。如需詳細資訊，請參閱AWS PCS 的 Amazon EC2 Spot 角色。

在 AWS PCS 中建立運算節點群組

您可以使用 AWS Management Console 或 建立運算節點群組 AWS CLI。

AWS Management Console

使用主控台建立運算節點群組

1. 開啟 AWS PCS 主控台。
2. 選取您要建立運算節點群組的叢集。導覽至運算節點群組，然後選擇建立。
3. 在運算節點群組設定區段中，提供節點群組的名稱。名稱只能包含區分大小寫的英數字元和連字號。它必須以字母字元開頭，且長度不可超過 25 個字元。名稱在叢集中必須是唯一的。
4. 在運算組態下，輸入或選取這些值：

   a. EC2 啟動範本 – 選取要用於此節點群組的自訂啟動範本。啟動範本可用來自訂網路設定，例如子網路、安全群組、監控組態和執行個體層級儲存。如果您沒有準備好啟動範本，請參閱 以搭配 AWS PCS 使用 Amazon EC2 啟動範本了解如何建立範本。

   > **Important**
   >
   > AWS PCS 會為每個運算節點群組建立受管啟動範本。這些名稱為 `pcs-identifier-do-not-delete`。請勿在建立或更新運算節點群組時選取這些節點群組，否則節點群組將無法正常運作。

   b. EC2 啟動範本版本 – 您必須選取自訂啟動範本的版本。如果您稍後變更版本，則必須更新運算節點群組，以偵測啟動範本中的變更。如需詳細資訊，請參閱更新 AWS PCS 運算節點群組。

   c. AMI ID – 如果您的啟動範本不包含 AMI ID，或您想要覆寫啟動範本中的值，請在此處提供 AMI ID。請注意，用於節點群組的 AMI 必須與 AWS PCS 相容。您也可以選取 提供的範例 AMI AWS。如需此主題的詳細資訊，請參閱 AWS PCS 的 Amazon Machine Image AMIs)。

   d. IAM 執行個體描述檔 – 選擇節點群組的執行個體描述檔。執行個體描述檔會授予執行個體安全存取 AWS 資源和服務的權限。如果您沒有預備的設定檔，您可以選取建立基本設定

檔，讓 AWS PCS 使用最低政策為您建立設定檔，或參閱 平行運算服務的 IAM AWS 執行個體描述檔。

e. 子網路 – 在部署 AWS PCS 叢集的 VPC 中選擇一或多個子網路。如果您選擇多個子網路，則節點之間無法使用 EFA 通訊，而且不同子網路中節點之間的通訊延遲可能會增加。請確定您在此處指定的子網路符合您在 EC2 啟動範本中定義的任何子網路。

f. 執行個體 – 選擇一或多個執行個體類型，以滿足節點群組中的擴展請求。所有執行個體類型都必須具有相同的處理器架構 (x86_64 或 arm64) 和 vCPUs 數量。如果執行個體具有 GPUs，則所有執行個體類型都必須具有相同數目的 GPUs。

g. 擴展組態 – 指定節點群組的執行個體數量下限和上限。您可以定義靜態組態，其中有固定數量的節點正在執行，或是動態組態，其中最多可以執行節點的計數上限。對於靜態組態，將最小值和最大值設定為相同，大於零個數字。對於動態組態，請將最小執行個體設定為零，最大執行個體設定為大於零的數字。 AWS PCS 不支援混合靜態和動態執行個體的運算節點群組。

5. （選用）在其他設定下，指定下列項目：

a. 購買選項 – 在 Spot 和隨需執行個體之間選取。

b. 配置策略 – 如果您已選取 Spot 購買選項，您可以指定在節點群組中啟動執行個體時，如何選擇 Spot 容量集區。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Spot 執行個體的配置策略。如果您已選取隨需購買選項，此選項不會有任何影響。

6. （選用）在Slurm自訂設定區段中，提供這些值：

a. 權重 – 此值會設定群組中節點的優先順序，以供排程之用。權重較低的節點具有較高的優先順序，而且單位是任意的。如需詳細資訊，請參閱 Slurm 文件中的權重。

b. 實際記憶體 – 此值會設定節點群組中節點上實際記憶體的大小（以 GB 為單位）。它旨在與 AWS PCS 中叢集Slurm組態中的 `CR_CPU_Memory`選項搭配使用。如需詳細資訊，請參閱 Slurm 文件中的 RealMemory。

7. （選用）在標籤下，將任何標籤新增至運算節點群組。

8. 選擇建立運算節點群組。當 AWS PCS 佈建節點群組`Creating`時，狀態欄位會顯示 。這可能需要幾分鐘的時間。

建議的下一個步驟

- 將節點群組新增至 AWS PCS 中的佇列，使其能夠處理任務。

AWS CLI

使用 建立運算節點群組 AWS CLI

使用下列命令建立您的佇列。執行命令之前，請執行下列替換：

1. 以 的 ID 取代*##*， AWS 區域 以在 中建立叢集，例如 us-east-1。
2. 以叢集的名稱或 取代 *my-cluster*clusterId。
3. 將 *my-node-group* 取代為運算節點群組的名稱。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 25 個字元。名稱在叢集中必須是唯一的。
4. 將 *subnet-ExampleID1* 取代為叢集 VPC 中的一或多個子網路 IDs。
5. 使用自訂啟動範本的 ID 取代 *lt-ExampleID1*。如果您沒有準備好，請參閱 以搭配 AWS PCS 使用 Amazon EC2 啟動範本了解如何建立。

   > **Important**
   >
   > AWS PCS 會為每個運算節點群組建立受管啟動範本。這些名稱為 pcs-*identifier*-do-not-delete。請勿在建立或更新運算節點群組時選取這些節點群組，否則節點群組將無法正常運作。

6. 將 *launch-template-version* 取代為特定的啟動範本版本。 AWS PCS 會將您的節點群組與該特定版本的啟動範本建立關聯。
7. 使用 IAM 執行個體描述檔的 ARN 取代 *arn#InstanceProfile*。如果您沒有準備，請參閱 搭配 AWS PCS 使用 Amazon EC2 啟動範本 以取得指引。
8. 以整數值取代 *min-instances* 和 *max-instances*。您可以定義靜態組態，其中有固定數量的節點正在執行，或是動態組態，其中最多可以執行節點的計數上限。對於靜態組態，將最小值和最大值設定為相同，大於零個數字。對於動態組態，將最小執行個體設定為零，將最大執行個體設定為大於零的數字。 AWS PCS 不支援混合靜態和動態執行個體的運算節點群組。
9. 將 *t3.large* 取代為另一個執行個體類型。您可以指定instanceType設定清單來新增更多執行個體類型。例如，*--instance-configs instanceType=c6i.16xlarge instanceType=c6a.16xlarge*。所有執行個體類型都必須具有相同的處理器架構 (x86_64 或 arm64) 和 vCPUs 數量。如果執行個體具有 GPUs，則所有執行個體類型都必須具有相同數目的 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
```

有幾個選用的組態設定可以新增至`create-compute-node-group`命令。

- 您可以指定自訂啟動範本`--amiId`是否不包含 AMI 的參考，或是您想要覆寫該值。請注意，用於節點群組的 AMI 必須與 AWS PCS 相容。您也可以選取 提供的範例 AMI AWS。如需此主題的詳細資訊，請參閱 AWS PCS 的 Amazon Machine Image AMIs)。
- 您可以使用 在隨需 (ONDEMAND) 和 Spot (SPOT) 執行個體之間進行選取`--purchase-option`。隨需是預設值。如果您選擇 Spot 執行個體，您也可以使用 `--allocation-strategy` 定義 AWS PCS 在節點群組中啟動執行個體時如何選擇 Spot 容量集區。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Spot 執行個體的配置策略。
- 您可以使用 為節點群組中的節點提供Slurm組態選項`--slurm-configuration`。您可以設定權重 （排程優先順序） 和實際記憶體。權重較低的節點具有較高的優先順序，而且單位是任意的。如需詳細資訊，請參閱 Slurm 文件中的權重。實際記憶體會設定節點群組中節點上實際記憶體的大小 （以 GB 為單位）。其旨在與Slurm組態中 AWS PCS 中叢集的 `CR_CPU_Memory`選項搭配使用。如需詳細資訊，請參閱 Slurm 文件中的 RealMemory。

> **Important**
>
> 建立運算節點群組可能需要幾分鐘的時間。

您可以使用下列命令查詢節點群組的狀態。在節點群組的狀態達到 之前，您將無法將節點群組與佇列建立關聯`ACTIVE`。

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

更新 AWS PCS 運算節點群組

本主題提供可用選項的概觀，並說明更新 AWS PCS 運算節點群組時應考量的事項。

更新 AWS PCS 運算節點群組的選項

更新 AWS PCS 運算節點群組可讓您變更 AWS PCS 啟動之執行個體的屬性，以及這些執行個體如何啟動的規則。例如，您可以將節點群組執行個體的 AMI 取代為另一個已安裝不同軟體的 AMI。或者，您可以更新安全群組，以變更傳入或傳出網路連線。您也可以變更擴展組態，甚至將偏好的購買選項變更為 Spot 執行個體或從 Spot 執行個體變更。

下列節點群組設定無法在建立後變更：

- 名稱
- 執行個體

更新 AWS PCS 運算節點群組時的考量事項

運算節點群組定義 EC2 執行個體，用於處理任務、提供互動式 shell 存取和其他任務。它們通常與一或多個 AWS PCS 佇列相關聯。當您更新運算節點群組以變更其行為（或其節點的行為）時，請考慮下列事項：

- 當運算節點群組狀態從更新到作用中時，運算節點群組屬性的變更就會生效。新的執行個體會以更新的屬性啟動。
- 不會影響特定節點組態的更新不會影響執行中的節點。例如，新增子網路並變更配置策略。
- 如果您更新運算節點群組的啟動範本，則必須更新運算節點群組才能使用新版本。
- 若要從運算節點群組中的節點新增或移除安全群組，請編輯其啟動範本並更新運算節點群組。使用更新的安全群組集啟動新的執行個體。
- 如果您直接編輯運算節點群組使用的安全群組，它會立即影響執行中和未來的執行個體。
- 如果您從運算節點群組使用的 IAM 執行個體描述檔新增或移除許可，則會立即影響執行中和未來的執行個體。
- 若要變更運算節點群組執行個體使用的 AMI，請更新運算節點群組（或其啟動範本）以使用新的 AMI，並等待 AWS PCS 取代執行個體。
- AWS PCS 會在節點群組更新操作後取代節點群組中的現有執行個體。如果節點上有任務正在執行，則允許在 AWS PCS 取代節點之前完成這些任務。互動式使用者程序（例如登入節點執行個體）會終止。當 AWS PCS 標記執行個體進行替換`Active`時，節點群組狀態會傳回 ，但實際替換會在執行個體閒置時發生。
- 如果您減少運算節點群組中允許的執行個體數量上限， AWS PCS 會從 Slurm 移除節點，以符合新的上限。 AWS PCS 會終止與已移除 Slurm 節點相關聯的執行中執行個體。已移除節點上的執行中任務會失敗並返回其佇列。

- AWS PCS 會為每個運算節點群組建立受管啟動範本。它們名為 `pcs-identifier-do-not-delete`。請勿在建立或更新運算節點群組時選取它們，否則節點群組將無法正常運作。
- 如果您更新運算節點群組以使用 Spot 作為其購買選項，則必須在帳戶中擁有 AWSServiceRoleForEC2Spot 服務連結角色。如需詳細資訊，請參閱AWS PCS 的 Amazon EC2 Spot 角色。

更新 AWS PCS 運算節點群組

您可以使用 AWS 管理主控台或 AWS CLI 更新節點群組。

AWS Management Console

更新運算節點群組

1. 在 開啟 AWS PCS 主控台 `https://console.aws.amazon.com/pcs/home#/clusters`
2. 選取您要更新運算節點群組的叢集。
3. 導覽至運算節點群組，前往您要更新的節點群組，然後選取編輯。
4. 在運算組態、其他設定和Slurm自訂設定區段中，更新任何值，除了：
   - 執行個體 – 您無法變更運算節點群組中的執行個體。
5. 選擇更新。套用變更時，狀態欄位會顯示更新。

> **Important**
> 運算節點群組更新可能需要幾分鐘的時間。

AWS CLI

更新運算節點群組

1. 使用下列命令更新您的運算節點群組。執行命令之前，請執行下列替換：
   a. 將 *region-code* 取代為您要建立叢集的 AWS 區域。
   b. 將 *my-node-group* 取代computeNodeGroupId為運算節點群組的名稱或 。
   c. 以clusterId叢集的名稱或 取代 *my-cluster*。

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

2. 更新 以外的任何節點群組參數--instance-configs。例如，若要設定新的 AMI ID，請傳遞 *my-custom-ami-id* 以您選擇的 AMI 取代--amiId my-custom-ami-id。

> **Important**
>
> 更新運算節點群組可能需要幾分鐘的時間。

您可以使用下列命令查詢節點群組的狀態。

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

在 AWS PCS 中刪除運算節點群組

本主題提供可用選項的概觀，並說明在 AWS PCS 中刪除運算節點群組時應考量的事項。

刪除運算節點群組時的考量事項

運算節點群組定義 EC2 執行個體，用於處理任務、提供互動式 shell 存取和其他任務。它們通常與一或多個 AWS PCS 佇列相關聯。刪除運算節點群組之前，請考慮下列事項：

- 運算節點群組啟動的任何 EC2 執行個體都將終止。這將取消在這些執行個體上執行的任務，並終止執行中的互動式程序。
- 您必須先取消運算節點群組與所有佇列的關聯，才能將其刪除。如需詳細資訊，請參閱更新 AWS PCS 佇列。

刪除運算節點群組

您可以使用 AWS Management Console 或 AWS CLI 來刪除運算節點群組。

AWS Management Console

刪除運算節點群組

1. 開啟 AWS PCS 主控台。
2. 選取運算節點群組的叢集。
3. 導覽至運算節點群組，然後選取要刪除的運算節點群組。
4. 選擇 刪除 。
5. 狀態欄位會顯示 `Deleting`。此需要幾分鐘的時間來完成。

> **Note**
>
> 您可以使用排程器原生的命令來確認運算節點群組已刪除。例如，針對 Slurm 使用 `squeue sinfo`或 。

AWS CLI

刪除運算節點群組

- 使用以下命令刪除運算節點群組，並取代這些節點：
  - AWS 區域 將 *region-code* 取代為您的叢集所在的 。
  - 將 *my-node-group* 取代為運算節點群組的名稱或 ID。
  - 以叢集的名稱或 ID 取代 *my-cluster*。

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

刪除運算節點群組可能需要幾分鐘的時間。

> **Note**
>
> 您可以使用排程器原生的命令來確認運算節點群組已刪除。例如，針對 Slurm 使用 `squeue sinfo`或 。

在 AWS PCS 中取得運算節點群組詳細資訊

您可以使用 AWS Management Console 或 AWS CLI 來取得運算節點群組的詳細資訊，例如其運算節點群組 ID、Amazon Resource Name (ARN) 和 Amazon Machine Image (AMI) ID。這些詳細資訊通常是 AWS PCS API 動作和組態的必要值。

AWS Management Console

取得運算節點群組詳細資訊

1. 開啟 AWS PCS 主控台。
2. 選取 叢集。
3. 選擇運算節點群組。
4. 從清單窗格中選擇運算節點群組。

AWS CLI

取得運算節點群組詳細資訊

1. 使用 ListClusters API 動作來尋找您的叢集名稱或 ID。

```
aws pcs list-clusters
```

輸出範例：

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

2. 使用 ListComputeNodeGroups API 動作列出叢集中的運算節點群組。

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

呼叫範例：

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

輸出範例：

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

3. 使用 GetComputeNodeGroup API 動作以取得運算節點群組的其他詳細資訊。

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

呼叫範例：

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

輸出範例：

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

在 AWS PCS 中尋找運算節點群組執行個體

每個 AWS PCS 運算節點群組都可以使用共用組態啟動 EC2 執行個體。您可以使用 EC2 標籤，在 AWS Management Console 或 中的運算節點群組中尋找執行個體 AWS CLI。

AWS Management Console

尋找您的運算節點群組執行個體

1. 開啟 AWS PCS 主控台。
2. 選取 叢集。
3. 選擇運算節點群組。
4. 尋找您建立之登入節點群組的 ID。
5. 導覽至 EC2 主控台，然後選擇執行個體。
6. 使用下列標籤搜尋執行個體。將 *node-group-id* 取代為您運算節點群組的 ID （而非名稱）。

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

7. （選用）您可以在搜尋欄位中變更執行個體狀態的值，以尋找正在設定或最近終止的執行個體。
8. 在已標記的執行個體清單中尋找每個執行個體的執行個體 ID 和 IP 地址。

AWS CLI

若要尋找節點群組執行個體，請使用下列命令。執行命令之前，請進行下列取代：

- *region-code* 將 取代為叢集 AWS 區域 的 。範例：us-east-1
- *node-group-id* 將 取代為您運算節點群組的 ID （而非名稱）。若要尋找運算節點群組的 ID，請參閱 在 AWS PCS 中取得運算節點群組詳細資訊。
- running 將 取代為其他執行個體狀態，例如 pending或 terminated，以尋找其他狀態的 EC2 執行個體。

```
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}
```

此命令會傳回如下輸出：PublicIP null 如果執行個體位於私有子網路中，則 的值為 。

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

> **Note**
>
> 如果您預期describe-instances傳回大量執行個體，您必須針對多個頁面使用選項。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud API 參考》中的DescribeInstances。

搭配 AWS PCS 使用 Amazon EC2 啟動範本

在 Amazon EC2 中，啟動範本可以存放一組偏好設定，這樣您就不必在啟動執行個體時個別指定。AWS PCS 將啟動範本納入為設定運算節點群組的彈性方式。當您建立節點群組時，您會提供啟動範本。 AWS PCS 會從中建立衍生的啟動範本，其中包含轉換，以協助確保其能與 服務搭配使用。

了解撰寫自訂啟動範本時有哪些選項和考量事項，可協助您撰寫一個以搭配 AWS PCS 使用的選項和考量事項。如需啟動範本的詳細資訊，請參閱《Amazon EC2 使用者指南》中的從啟動範本啟動執行個體。

主題

AWS PCS 啟動範本概觀

您可以在 EC2 啟動範本中包含超過 30 個參數，控制執行個體的許多設定方式。大多數都與 AWS PCS 完全相容，但有一些例外狀況。

AWS PCS 會忽略 EC2 啟動範本的下列參數，因為這些服務必須直接管理這些屬性：

- 執行個體類型/指定執行個體類型屬性 (`InstanceRequirements`) – AWS PCS 不支援屬性型執行個體選取。
- 執行個體類型 (`InstanceType`) – 建立節點群組時指定執行個體類型。
- 進階詳細資訊/IAM 執行個體描述檔 (`IamInstanceProfile`) – 您在建立或更新節點群組時提供此描述檔。
- 進階詳細資訊/停用 API 終止 (`DisableApiTermination`) – AWS PCS 必須控制其啟動的節點群組執行個體生命週期。
- 進階詳細資訊/停用 API 停止 (`DisableApiStop`) – AWS PCS 必須控制其啟動的節點群組執行個體生命週期。
- 進階詳細資訊/停止 – 休眠行為 (`HibernationOptions`) – AWS PCS 不支援執行個體休眠。

- 進階詳細資訊/彈性 GPU (ElasticGpuSpecifications) – Amazon Elastic Graphics 已於 2024 年 1 月 8 日終止服務。
- 進階詳細資訊/彈性推論 (ElasticInferenceAccelerators) – Amazon Elastic Inference 不再提供給新客戶。
- AAdvanced 詳細資訊/指定 CPU 選項/每個核心執行緒 (ThreadsPerCore) – AWS PCS 將每個核心的執行緒數目設定為 1。

這些參數有特殊需求，可支援與 AWS PCS 的相容性：

- 使用者資料 (UserData) – 必須為分段編碼。請參閱 使用 AWS PCS 的 Amazon EC2 使用者資料。
- 應用程式和作業系統映像 (ImageId) – 您可以包含此項目。不過，如果您在建立或更新節點群組時指定 AMI ID，則會覆寫啟動範本中的值。您提供的 AMI 必須與 AWS PCS 相容。如需詳細資訊，請參閱「AWS PCS 的 Amazon Machine Image AMIs)。
- 網路設定/防火牆 （ 安全群組 ） (SecurityGroups) – 無法在 AWS PCS 啟動範本中設定安全群組名稱的清單。您可以設定安全群組 IDs(SecurityGroupIds) 的清單，除非您在啟動範本中定義網路介面。然後，您必須為每個界面指定安全群組 IDs。如需詳細資訊，請參閱AWS PCS 中的安全群組。
- 網路設定/進階網路組態 (NetworkInterfaces) – 如果您使用 EC2 執行個體搭配單一網路卡，而且不需要任何專門的網路組態， AWS PCS 可以為您設定執行個體聯網。若要設定多個網路卡或在您的執行個體上啟用 Elastic Fabric Adapter，請使用 NetworkInterfaces。每個網路界面都必須有下的安全群組 IDs 清單Groups。如需詳細資訊，請參閱AWS PCS 中的多個網路介面。
- 進階詳細資訊/容量保留 (CapacityReservationSpecification) – 可以設定此項目，但在使用 AWS PCS CapacityReservationId時無法參考特定項目。不過，您可以參考容量保留群組，其中該群組包含一或多個容量保留。如需詳細資訊，請參閱AWS PCS 中的容量保留。

建立基本的啟動範本

您可以使用 AWS Management Console 或 建立啟動範本 AWS CLI。

AWS Management Console

建立啟動範本

1. 開啟 Amazon EC2 主控台，然後選取啟動範本。
2. 選擇 Create launch template (建立啟動範本)。

3. 在啟動範本名稱和描述下，輸入啟動範本名稱的唯一、特殊的名稱
4. 在金鑰對名稱的金鑰對（登入）下，選取將用於登入 AWS PCS 管理之 EC2 執行個體的 SSH 金鑰對。此為選用操作，但建議您採用。
5. 在網路設定下，接著防火牆（安全群組），選擇要連接至網路介面的安全群組。啟動範本中的所有安全群組都必須來自您的 AWS PCS 叢集 VPC。至少，選擇：
   - 允許與 AWS PCS 叢集通訊的安全群組
   - 允許 AWS PCS 啟動之 EC2 執行個體之間的通訊的安全群組
   - （選用）允許傳入 SSH 存取互動式執行個體的安全群組
   - （選用）允許運算節點對網際網路進行傳出連線的安全群組
   - （選用）允許存取網路資源的安全群組（例如共用檔案系統或資料庫伺服器）。
6. 您可以在 Amazon EC2 主控台的啟動範本下存取新的啟動範本 ID。啟動範本 ID 會有表單 `lt-0123456789abcdef01`。

建議的下一個步驟

- 使用新的啟動範本來建立或更新 AWS PCS 運算節點群組。

AWS CLI

建立啟動範本

使用下列命令建立您的啟動範本。

- 執行命令之前，請執行下列替換：
  a. 將 *region-code* 取代為您使用 AWS PCS AWS 區域 的
  b. 將 *my-launch-template-name* 取代為範本的名稱。它對於 必須是唯一的 AWS 帳戶 ，而且 AWS 區域 您正在使用。
  c. 將 *my-ssh-key-name* 取代為您偏好的 SSH 金鑰名稱。
  d. 將 *sg-ExampleID1* 和 *sg-ExampleID2* 取代為安全群組 IDs，允許 EC2 執行個體與排程器之間的通訊，以及 EC2 執行個體之間的通訊。如果您只有一個啟用所有此流量的安全群組，您可以移除 `sg-ExampleID2`及其前面的逗號字元。您也可以新增更多安全群組 IDs。您在啟動範本中包含的所有安全群組都必須來自 AWS PCS 叢集 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 將輸出類似以下內容的文字。啟動範本 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"
    }
}
```

建議的下一個步驟

- 使用新的啟動範本來建立或更新 AWS PCS 運算節點群組。

使用 AWS PCS 的 Amazon EC2 使用者資料

您可以在執行個體啟動時cloud-init執行的啟動範本中提供 EC2 使用者資料。具有內容類型的使用者資料區塊會在執行個體向 AWS PCS API 註冊之前cloud-config執行，而具有內容類型的使用者資料區塊會在註冊完成後執行，但在 Slurm 協助程式啟動之前text/x-shellscript執行。如需內容類型的詳細資訊，請參閱 cloud-init 文件。

我們的使用者資料可以執行常見的組態案例，包括但不限於下列項目：

- 包含使用者或群組
- 安裝套件
- 建立分割區和檔案系統
- 掛載網路檔案系統

啟動範本中的使用者資料必須是 MIME 分段封存格式。這是因為您的使用者資料會與其他 AWS PCS 使用者資料合併，這是設定節點群組中節點所需的資料。您可以將多個使用者資料區塊組合在一起成為單一 MIME 分段檔案。

MIME 分段檔案包含下列元件：

- 內容類型和部分邊界宣告：`Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- MIME 版本宣告：`MIME-Version: 1.0`
- 一或多個使用者資料區塊，其中包含下列元件：
  - 發出使用者資料區塊開頭訊號的開啟界限：`--==BOUNDARY==`。您必須在此邊界之前保留該行空白。
  - 區塊的內容類型宣告：`Content-Type: text/cloud-config; charset="us-ascii"`或`Content-Type: text/x-shellscript; charset="us-ascii"`。您必須在內容類型宣告後保留該行空白。
  - 使用者資料的內容，例如 shell 命令或`cloud-config`指令的清單。
- 發出 MIME 分段檔案結尾訊號的結束界限：`--==BOUNDARY==--`。您必須在結束界限之前將行保留空白。

> **Note**
>
> 如果您在 Amazon EC2 主控台中將使用者資料新增至啟動範本，您可以將其以純文字形式貼入。或者，您可以從檔案上傳。如果您使用 AWS CLI 或 AWS 開發套件，則必須先對使用者資料進行 base64 編碼，並在呼叫 CreateLaunchTemplate 時將該字串提交為 `UserData` 參數的值，如此 JSON 檔案所示。

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

範例

範例：從套件儲存庫安裝 AWS PCS 的軟體

在啟動範本"userData"中提供此指令碼做為 的值。如需詳細資訊，請參閱使用 AWS PCS 的 Amazon EC2 使用者資料。

此指令碼使用 cloud-config，在啟動時在節點群組執行個體上安裝軟體套件。如需詳細資訊，請參閱 Cloud-init 文件中的使用者資料格式。此範例會安裝 `curl`和 `llvm`。

> **Note**
>
> 您的執行個體必須能夠連線到其設定的套件儲存庫。

```
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==--
```

範例：從 S3 儲存貯體執行 AWS PCS 的其他指令碼

在啟動範本"userData"中提供此指令碼做為 的值。如需詳細資訊，請參閱使用 AWS PCS 的 Amazon EC2 使用者資料。

下列使用者資料指令碼使用 cloud-config 從 S3 儲存貯體匯入指令碼，並在啟動時在節點群組執行個體上執行。如需詳細資訊，請參閱 Cloud-init 文件中的使用者資料格式。

將下列值取代為您自己的詳細資訊：

- *amzn-s3-demo-bucket* – 您的帳戶可讀取的 S3 儲存貯體名稱。
- *object-key* – 要匯入之指令碼的 S3 物件金鑰。這包括指令碼的名稱及其在儲存貯體資料夾結構中的位置。例如：scripts/script.sh。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用資料夾在 Amazon S3 主控台中組織物件。
- *shell* – 用來執行指令碼的 Linux shell，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==--
```

節點群組的 IAM 執行個體描述檔必須具有 儲存貯體的存取權。下列 IAM 政策是上述使用者資料指令碼中儲存貯體的範例。

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

```
    ]
}
```

範例：設定 AWS PCS 的全域環境變數

在啟動範本"userData"中提供此指令碼做為 的值。如需詳細資訊，請參閱使用 AWS PCS 的 Amazon EC2 使用者資料。

下列範例使用 在節點群組執行個體上/etc/profile.d設定全域變數。

```
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==--
```

範例：使用 EFS 檔案系統做為 AWS PCS 的共用主目錄

在啟動範本"userData"中提供此指令碼做為 的值。如需詳細資訊，請參閱使用 AWS PCS 的 Amazon EC2 使用者資料。

此範例延伸了 中的範例 EFS 掛載搭配 AWS PCS 使用網路檔案系統，以實作共用的主目錄。在掛載 EFS 檔案系統之前，會備份 /home 的內容。掛載完成後，內容會快速複製到共用儲存體上的適當位置。

使用您自己的詳細資訊取代此指令碼中的下列值：

- *`/mount-point-directory`* – 您想要掛載 EFS 檔案系統的執行個體路徑。
- *`filesystem-id`* – EFS 檔案系統的檔案系統 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==--
```

範例：啟用無密碼 SSH

您可以建置共用主目錄範例，使用 SSH 金鑰在叢集執行個體之間實作 SSH 連線。針對使用共用主檔案系統的每個使用者，執行類似下列的指令碼：

```
#!/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**
>
> 執行個體必須使用允許叢集節點之間 SSH 連線的安全群組。

AWS PCS 中的容量保留

您可以使用隨需容量保留或 Amazon EC2 EC2 容量，以確保在需要時有可用的必要運算容量。

> **Note**
>
> AWS PCS 支援隨需容量保留 (ODCR)，但目前不支援 ML 的容量區塊。

搭配 AWS PCS 使用 ODCRs

您可以選擇 AWS PCS 使用預留執行個體的方式。如果您建立開啟的 ODCR， AWS PCS 或您帳戶中的其他程序啟動的任何相符執行個體都會計入保留。使用目標 ODCR，只有以特定保留 ID 啟動的執行個體才會計入保留。對於時間敏感工作負載，目標 ODCRs 更常見。

您可以將 AWS PCS 運算節點群組新增至啟動範本，以使用目標 ODCR。以下是執行此操作的步驟：

1. 建立目標隨需容量保留 (ODCR)。
2. 將 ODCR 新增至容量保留群組。
3. 將容量保留群組與啟動範本建立關聯。
4. 建立或更新 AWS PCS 運算節點群組以使用啟動範本。

範例：保留並使用具有目標 ODCR 的 hpc6a.48xlarge 執行個體

此範例命令會為 32 hpc6a.48xlarge 執行個體建立目標 ODCR。若要在置放群組中啟動預留執行個體，請將 `--placement-group-arn`新增至 命令。您可以使用 `--end-date`和 定義停止日期`--end-date-type`，否則保留會持續到手動終止為止。

```
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
```

此命令的結果將是新 ODCR 的 ARN。若要將 ODCR 與 AWS PCS 搭配使用，則必須將其新增至容量保留群組。這是因為 AWS PCS 不支援個別 ODCRs。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的容量保留群組。

以下是如何將 ODCR 新增至名為 的容量保留群組。 `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
```

建立 ODCR 並新增至容量保留群組後，即可將其新增至啟動範本，以連接至 AWS PCS 運算節點群組。以下是參考容量保留群組的範例啟動範本。

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

最後，建立或更新 AWS PCS 運算節點群組以使用 hpc6a.48xlarge 執行個體，並使用在其容量保留群組中參考 ODCR 的啟動範本。對於靜態節點群組，將最小和最大執行個體設定為保留的大小 (32)。對於動態節點群組，將最小執行個體設定為 0，並將最大執行個體設定為保留大小。

此範例是為一個運算節點群組佈建的單一 ODCR 的簡單實作。但是， AWS PCS 支援許多其他設計。例如，您可以在多個運算節點群組之間分割大型 ODCR 或容量保留群組。或者，您可以使用另一個 AWS 帳戶建立並與您共用的 ODCRs。金鑰限制條件是 ODCRs 一律必須包含在容量保留群組中。

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 ML 的隨需容量預留和容量區塊。

實用的啟動範本參數

本節說明一些啟動範本參數，這些參數可能對 AWS PCS 廣泛有用。

開啟詳細的 CloudWatch 監控

您可以使用啟動範本參數，以較短的間隔啟用 CloudWatch 指標集合。

AWS Management Console

在建立或編輯啟動範本的主控台頁面上，此選項位於進階詳細資訊區段下。將詳細 CloudWatch 監控設定為啟用。

YAML

```
Monitoring:
```

```
    Enabled: True
```

JSON

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

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的啟用或停用執行個體的詳細監控。

執行個體中繼資料服務第 2 版 (IMDS v2)

將 IMDS v2 與 EC2 執行個體搭配使用可提供顯著的安全增強功能，並有助於降低在 AWS 環境中存取執行個體中繼資料的潛在風險。

AWS Management Console

在建立或編輯啟動範本的主控台頁面上，此選項位於進階詳細資訊區段下。將已啟用可存取的中繼資料、僅限 V2 的中繼資料版本 （需要金鑰），以及中繼資料回應跳轉限制設為 4。 V2

YAML

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

JSON

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

.

AWS PCS 佇列

AWS PCS 佇列是排程器原生實作工作佇列的輕量抽象。如果是 Slurm， AWS PCS 佇列相當於 Slurm 分割區。

使用者將任務提交到他們所在的佇列，直到可以排程在一或多個運算節點群組提供的節點上執行。AWS PCS 叢集可以有多個任務佇列。例如，您可以建立使用 Amazon EC2 隨需執行個體進行高優先順序任務的佇列，以及使用 Amazon EC2 Spot 執行個體進行低優先順序任務的另一個佇列。

主題

在 AWS PCS 中建立佇列

本主題提供可用選項的概觀，並說明您在 AWS PCS 中建立佇列時應考量的事項。

先決條件

- AWS PCS 叢集 - 只能建立與特定 AWS PCS 叢集相關聯的佇列。
- 一或多個 AWS PCS 運算節點群組 - 佇列必須與至少一個 AWS PCS 運算節點群組相關聯。

在 AWS PCS 中建立佇列

您可以使用 AWS Management Console 或 建立佇列 AWS CLI。

AWS Management Console

使用主控台建立佇列

1. 開啟 AWS PCS 主控台。
2. 選取佇列的叢集。導覽至佇列，然後選擇建立佇列。
3. 在佇列組態區段中，提供下列值：
    a. 佇列名稱 – 佇列的名稱。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 25 個字元。名稱在叢集中必須是唯一的。

b. 運算節點群組 – 選取一或多個運算節點群組來服務此佇列。運算節點群組可以與多個佇列建立關聯。

4. （選用）在標籤下，將任何標籤新增至 AWS PCS 佇列
5. 選擇建立佇列。當 AWS PCS 建立佇列時，狀態欄位會顯示建立。建立佇列可能需要幾分鐘的時間。

建議的下一個步驟

- 將任務提交至您的新佇列。

AWS CLI

使用 建立佇列 AWS CLI

使用下列命令來建立佇列。進行下列取代：

1. 將###取代為叢集 AWS 的區域。例如：us-east-1。
2. 將 *my-queue* 取代為佇列的名稱。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 25 個字元。名稱在叢集中必須是唯一的。
3. 以叢集的名稱或 ID 取代 *my-cluster*。
4. 將 *compute-node-group-id* 取代為運算節點群組的 ID，以服務佇列。例如：pcs_abcdef12345。

> **Note**
>
> 建立佇列時，您必須提供運算節點群組的 ID，而不是其名稱。

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

建立佇列可能需要幾分鐘的時間。您可以使用下列命令查詢佇列的狀態。在佇列的狀態達到 之前，您將無法將任務提交至佇列ACTIVE。

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

建議的下一個步驟

- 將任務提交到您的新佇列

更新 AWS PCS 佇列

本主題提供可用選項的概觀，並說明更新 AWS PCS 佇列時應考量的事項。

更新 AWS PCS 佇列時的考量事項

佇列更新不會影響執行中的任務，但叢集在更新佇列時可能無法接受新任務。

更新 AWS PCS 佇列

您可以使用 AWS Management Console 或 AWS CLI 來更新佇列。

AWS Management Console

更新佇列

1. 在 開啟 AWS PCS 主控台 `https://console.aws.amazon.com/pcs/home#/clusters`
2. 選取您要更新佇列的叢集。
3. 導覽至佇列，前往要更新的佇列，然後選取編輯。
4. 在佇列組態區段中，更新下列任何值：
   - 節點群組 – 新增或移除運算節點群組與佇列的關聯。
   - 標籤 – 新增或移除佇列的標籤。
5. 選擇更新。套用變更時，狀態欄位會顯示更新。

> **Important**
> 佇列更新可能需要幾分鐘的時間。

AWS CLI

更新佇列

1. 使用下列命令更新您的佇列。執行命令之前，請執行下列替換：

   a. 將 *region-code* 取代為您要在其中建立叢集 AWS 區域 的 。

   b. 將 *my-queue* 取代computeNodeGroupId為佇列的名稱或 。

   c. 將 *my-cluster* 取代為叢集的名稱或 clusterId 。

   d. 若要變更運算節點群組關聯，請提供 的更新清單--compute-node-group-configurations。

      - 例如，若要新增第二個運算節點群組computeNodeGroupExampleID2：

```
--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. 更新佇列可能需要幾分鐘的時間。您可以使用下列命令查詢佇列的狀態。在任務的狀態達到 之前，您將無法將任務提交至佇列ACTIVE。

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

建議的後續步驟

- 將任務提交至您更新的佇列。

在 AWS PCS 中刪除佇列

本主題提供如何在 AWS PCS 中刪除佇列的概觀。

刪除佇列時的考量事項

- 如果佇列中有正在執行的任務，則在刪除佇列時，排程器將會終止這些任務。佇列中的待定任務將會取消。考慮等待佇列中的任務完成，或使用排程器的原生命令手動停止/取消任務 （ 例如 `scancel` Slurm)。

刪除佇列

您可以使用 AWS Management Console 或 AWS CLI 來刪除佇列。

AWS Management Console

刪除佇列

1. 開啟 AWS PCS 主控台。
2. 選取佇列的叢集。
3. 導覽至佇列，然後選取要刪除的佇列。
4. 選擇 刪除 。
5. 狀態欄位會顯示 `Deleting`。此需要幾分鐘的時間來完成。

> **Note**
> 您可以使用排程器原生的命令來確認佇列已刪除。例如，針對 Slurm 使用 `squeue` `sinfo`或 。

AWS CLI

刪除佇列

- 使用下列命令來刪除佇列，並取代這些佇列：
  - AWS 區域 將 *`region-code`* 取代為您的叢集所在的 。
  - 以佇列的名稱或 ID 取代 *`my-queue`*。
  - 以叢集的名稱或 ID 取代 *`my-cluster`*。

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

刪除佇列可能需要幾分鐘的時間。

> **Note**
>
> 您可以使用排程器原生的命令來確認佇列已刪除。例如，針對 Slurm 使用 `squeue sinfo`或 。

AWS PCS 登入節點

AWS PCS 叢集通常需要至少 1 個登入節點，以支援互動式存取和任務管理。一種實現此目標的方法，是使用為登入節點功能設定的靜態 AWS PCS 運算節點群組。您也可以設定獨立 EC2 執行個體做為登入節點。

主題

使用 AWS PCS 運算節點群組提供登入節點

本主題提供建議的組態選項概觀，並說明當您使用 AWS PCS 運算節點群組來持續提供叢集的互動式存取時，應考量的事項。

建立登入節點的 AWS PCS 運算節點群組

在操作上，這與建立一般運算節點群組沒有太大不同。不過，有一些關鍵組態選擇：

- 設定運算節點群組中至少一個 EC2 執行個體的靜態擴展組態。
- 選擇隨需購買選項，以避免回收執行個體 (s)。
- 選擇運算節點群組的資訊名稱，例如登入。
- 如果您希望登入節點執行個體可在 VPC 外部存取，請考慮使用公有子網路。
- 如果您想要允許 SSH 存取，啟動範本將需要一個安全群組，該安全群組會將 SSH 連接埠公開到您選擇的 IP 地址。
- IAM 執行個體描述檔應該只有您希望最終使用者擁有的 AWS 許可。如需詳細資訊，請參閱 平行運算服務的 IAM AWS 執行個體描述檔。
- 考慮允許 AWS Systems Manager Session Manager 管理您的登入執行個體。
- 考慮限制只有管理使用者才能存取執行個體 AWS 登入資料
- 選取較一般運算節點群組便宜的執行個體類型，因為登入節點會持續執行。
- 使用與其他運算節點群組相同的 （或衍生） AMI，以協助確保所有執行個體都安裝相同的軟體。如需自訂 AMIs的詳細資訊，請參閱 AWS PCS 的 Amazon Machine Image AMIs)
- 在登入節點上設定與運算執行個體相同的網路檔案系統 (Amazon EFS、Amazon FSx for Lustre 等） 掛載。如需詳細資訊，請參閱搭配 AWS PCS 使用網路檔案系統。

存取您的登入節點

一旦新的運算節點群組達到 ACTIVE 狀態，您就可以找到它已建立的 EC2 執行個體（並登入）。如需詳細資訊，請參閱在 AWS PCS 中尋找運算節點群組執行個體。

更新登入節點的 AWS PCS 運算節點群組

您可以使用 UpdateComputeNodeGroup 更新登入節點群組。作為節點群組更新程序的一部分，執行中的執行個體將被取代。請注意，這會中斷執行個體上任何作用中的使用者工作階段或程序。執行中或佇列的 Slurm 任務不會受到影響。如需詳細資訊，請參閱更新 AWS PCS 運算節點群組。

您也可以編輯運算節點群組使用的啟動範本。您必須使用 UpdateComputeNodeGroup 將更新的啟動範本套用至運算節點群組。在運算節點群組中啟動的新 EC2 執行個體會使用更新的啟動範本。如需詳細資訊，請參閱搭配 AWS PCS 使用 Amazon EC2 啟動範本。

刪除登入節點的 AWS PCS 運算節點群組

您可以使用 AWS PCS 中的刪除運算節點群組機制來更新登入節點群組。執行中的執行個體將在節點群組刪除時終止。請注意，這會中斷執行個體上任何作用中的使用者工作階段或程序。執行中或佇列的 Slurm 任務不會受到影響。如需詳細資訊，請參閱在 AWS PCS 中刪除運算節點群組。

使用獨立執行個體做為 AWS PCS 登入節點

您可以設定獨立的 EC2 執行個體來與 AWS PCS 叢集的 Slurm 排程器互動。這對於建立使用 AWS PCS 叢集但在 PCS 管理之外操作的登入節點、工作站或專用工作流程 AWS 管理主機非常有用。若要這樣做，每個獨立執行個體都必須：

1. 安裝相容的 Slurm 軟體版本。
2. 能夠連線至 AWS PCS 叢集的 Slurmctld 端點。
3. 使用 AWS PCS 叢集的端點和秘密正確設定 Slurm Auth 和 Cred Kiosk Daemon (`sackd`)。如需詳細資訊，請參閱 Slurm 文件中的封裝。

本教學課程可協助您設定連接至 AWS PCS 叢集的獨立執行個體。

內容

- 步驟 3 – 在執行個體上安裝 Slurm
- 步驟 4 – 擷取和存放叢集秘密
- 步驟 5 – 設定 AWS PCS 叢集的連線
- 步驟 6 – （選用） 測試連線

步驟 1 – 擷取目標 AWS PCS 叢集的地址和秘密

使用 AWS CLI 搭配接下來的 命令，擷取目標 AWS PCS 叢集的詳細資訊。執行命令之前，請執行下列替換：

- 將 *region-code* 取代為目標叢集執行 AWS 區域 所在的 。
- 將 *cluster-ident* 取代為目標叢集的名稱或識別符

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

命令會傳回類似此範例的輸出。

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

在此範例中，叢集 Slurm 控制器端點的 IP 地址為 ，`10.3.149.220`且正在連接埠 上執行`6817`。`secretArn` 將在後續步驟中使用 來擷取叢集秘密。稍後的步驟將使用 IP 地址和連接埠來設定`sackd`服務。

步驟 2 – 啟動 EC2 執行個體

啟動 EC2 執行個體

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。
3. （選用）在名稱和標籤區段中，提供執行個體的名稱，例如 `PCS-LoginNode`。該名稱將指派作為執行個體的資源標籤 (`Name=PCS-LoginNode`)。
4. 在應用程式和作業系統映像區段中，為 AWS PCS 支援的其中一個作業系統選取 AMI。如需詳細資訊，請參閱支援的作業系統。
5. 在執行個體類型區段中，選取支援的執行個體類型。如需詳細資訊，請參閱支援的執行個體類型。
6. 在金鑰對區段中，選取要用於執行個體的 SSH 金鑰對。
7. 在網路設定區段中：
    - 選擇編輯。
        i. 選取 AWS PCS 叢集的 VPC。

ii. 針對防火牆 (安全群組)，選擇選取現有的安全群組。

A. 選取允許執行個體與目標 AWS PCS 叢集 Slurm 控制器之間流量的安全群組。如需詳細資訊，請參閱安全群組需求和考量事項。

B. （選用）選取允許傳入 SSH 存取執行個體的安全群組。

8. 在儲存區段中，視需要設定儲存磁碟區。請務必設定足夠的空間來安裝應用程式和程式庫，以啟用您的使用案例。
9. 在進階下，選擇允許存取叢集秘密的 IAM 角色。如需詳細資訊，請參閱取得 Slurm 叢集秘密。
10. 在摘要窗格中，選擇啟動執行個體。

步驟 3 – 在執行個體上安裝 Slurm

當執行個體啟動並變成作用中時，請使用您偏好的機制連線到執行個體。使用 提供的 Slurm 安裝程式 AWS ，在執行個體上安裝 Slurm。如需詳細資訊，請參閱Slurm 安裝程式。

下載 Slurm 安裝程式，解壓縮，然後使用`installer.sh`指令碼安裝 Slurm。如需詳細資訊，請參閱步驟 3 – 安裝 Slurm。

步驟 4 – 擷取和存放叢集秘密

這些指示需要 AWS CLI。如需詳細資訊，請參閱《 第 2 版使用者指南》中的安裝或更新至最新版本的 AWS CLI AWS Command Line Interface 。

使用下列命令存放叢集秘密。

- 建立 Slurm 的組態目錄。

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

- 擷取、解碼和存放叢集秘密。執行此命令之前，請將 *region-code* 取代為目標叢集正在執行的區域，並將 *secret-arn* 取代為步驟 1 中`secretArn`擷取的 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**
>
> 在多使用者環境中，任何可存取執行個體的使用者，如果可以存取執行個體中繼資料服務 (IMDS)，就可能可以擷取叢集秘密。這反過來可以允許他們模擬其他使用者。考慮將 IMDS 的存取權限制為僅限根使用者或管理使用者。或者，請考慮使用不依賴執行個體描述檔的不同機制來擷取和設定秘密。

- 設定 Slurm 金鑰檔案的擁有權和許可。

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

> **Note**
>
> Slurm 金鑰必須由執行sackd服務的使用者和群組所擁有。

步驟 5 – 設定 AWS PCS 叢集的連線

若要建立 AWS PCS 叢集的連線，請依照下列步驟啟動 sackd 做為系統服務。

1. 使用下列命令設定 sackd服務的環境檔案。執行命令之前，請將 *ip-address* 和 *port* 取代為步驟 1 中從端點擷取的值。

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

2. 建立用於管理sackd程序systemd的服務檔案。

   ```
   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. 設定sackd服務檔案的擁有權。

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

4. 啟用 sackd服務。

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

5. 啟動 sackd 服務。

```
sudo systemctl start sackd
```

步驟 6 – （選用） 測試連線

確認sackd服務正在執行。範例輸出如下。如果有錯誤，通常會顯示在這裡。

```
[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
```

確認與叢集的連線正在使用 Slurm 用戶端命令，例如 sinfo和 squeue。以下是來自 的範例輸出sinfo。

```
[root@ip-10-3-27-112 ~]# /opt/aws/pcs/scheduler/slurm-24.11/bin/sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
all up infinite 4 idle~ compute-[1-4]
```

您也應該能夠提交任務。例如，類似此範例的命令會在叢集中的 1 個節點上啟動互動式任務。

```
/opt/aws/pcs/scheduler/slurm-24.11/bin/srun --nodes=1 -p all --pty bash -i
```

AWS PCS 網路

您的 AWS PCS 叢集是在 Amazon VPC 中建立。本章包含下列有關叢集排程器和節點聯網的主題。

除了選擇要在其中啟動執行個體的子網路之外，您必須使用 EC2 啟動範本來設定 AWS PCS 運算節點群組的網路。如需啟動範本的詳細資訊，請參閱 搭配 AWS PCS 使用 Amazon EC2 啟動範本。

主題

AWS PCS VPC 和子網路需求和考量事項

當您建立 AWS PCS 叢集時，您可以在該 VPC 中指定子網路 VPC。本主題概述您搭配叢集使用之 VPC 和子網路的 AWS PCS 特定需求和考量事項 (VPC)。如果您沒有 VPC 可與 AWS PCS 搭配使用，您可以使用 AWS提供的 AWS CloudFormation 範本建立 VPC。如需 VPCs的詳細資訊，請參閱《Amazon VPC 使用者指南》中的虛擬私有雲端 (VPC)。

VPC 要求和注意事項

當您建立叢集時，您指定的 VPC 必須符合下列要求和注意事項：

- VPC 必須擁有足夠數量的 IP 地址，可用於您要建立的叢集、任何節點和其他叢集資源。如需詳細資訊，請參閱《Amazon VPCs 使用者指南》中的 VPC 和子網路 IP 定址。
- VPC 必須具有 DNS 主機名稱和 DNS 解析支援。否則，節點無法註冊客戶叢集。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。
- VPC 可能需要使用 的 VPC AWS PrivateLink 端點，才能聯絡 AWS PCS API。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 將 VPC 連接到 服務 AWS PrivateLink。

> **Important**
>
> AWS PCS 不支援具有專用執行個體租用的 VPC。您用於 AWS PCS 的 VPC 必須使用`default`執行個體租用。您可以變更現有 VPC 的執行個體租用。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的變更 VPC 的執行個體租用。

子網需求和注意事項

當您建立 Slurm 叢集時， AWS PCS 會在您指定的子網路中建立彈性網路介面 (ENI)。此網路介面可啟用排程器控制器與客戶 VPC 之間的通訊。網路介面也可讓 Slurm 與客戶帳戶中部署的元件通訊。您只能在建立時指定叢集的子網路。

叢集的子網路要求

您在建立叢集時指定的子網路必須符合下列要求：

- 子網路必須至少有 1 個 IP 地址供 AWS PCS 使用。
- 子網路無法位於 AWS Outposts AWS Wavelength或 AWS Local Zone。
- 子網路可以是公有或私有。如果可能，我們建議您指定私有子網路。公有子網路是具有路由表的子網路，其中包含網際網路閘道的路由；私有子網路是具有路由表的子網路，不包含網際網路閘道的路由。

節點的子網路要求

您可以將節點和其他叢集資源部署到您在建立 AWS PCS 叢集時指定的子網路，以及相同 VPC 中的其他子網路。

您部署節點和叢集資源到 的任何子網路必須符合下列要求：

- 您必須確保子網路有足夠的可用 IP 地址來部署所有節點和叢集資源。
- 如果您打算將節點部署到公有子網路，該子網路必須自動指派 IPv4 公有地址。
- 如果您部署節點的子網路是私有子網路，且其路由表不包含網路地址轉譯 (NAT) 裝置 (IPv4) 的路由，請使用 AWS PrivateLink 將 VPC 端點新增至客戶 VPC。節點聯絡的所有 AWS 服務都需要 VPC 端點。唯一的必要端點是 AWS PCS 允許節點呼叫 `RegisterComputeNodeGroupInstance` API 動作。如需詳細資訊，請參閱 AWS PCS API 參考中的 RegisterComputeNodeGroupInstance。
- 公有或私有子網路狀態不會影響 AWS PCS；必要的端點必須可連線。

為您的 AWS PCS 叢集建立 VPC

您可以在 AWS 平行運算服務 (AWS PCS) 中為叢集建立 Amazon Virtual Private Cloud (Amazon VPC)。

使用 Amazon VPC 將 VPC 資源啟動至您定義的虛擬網路。此虛擬網路非常近似於您在自有資料中心內運作的傳統網路。但是，它帶來了使用 Amazon Web 服務的可擴展基礎架構的好處。我們建議您在部署生產 VPC 叢集之前，先徹底了解 Amazon VPC 服務。如需詳細資訊，請參閱 作者視覺化模式中的什麼是 Amazon VPC？。Amazon VPC 使用者指南。

PCS 叢集、節點和支援資源 （ 例如檔案系統和目錄服務 ） 會部署在您的 Amazon VPC 中。如果您想要將現有的 Amazon VPC 與 PCS AWS PCS VPC 和子網路需求和考量事項 搭配使用，則必須符合 中所述的要求。本主題說明如何使用 AWS提供的 AWS CloudFormation 範本建立符合 PCS 要求的 VPC。部署範本後，您可以檢視範本所建立的資源，以確切了解其建立的資源以及這些資源的組態。

先決條件

若要建立 Amazon VPC for PCS，您必須擁有必要的 IAM 許可才能建立 Amazon VPC 資源。這些資源包括 VPC、子網、安全群組、路由表和路由，以及網際網路和 NAT 閘道。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用公有子網路建立 VPC。 若要檢閱 Amazon EC2 的完整清單，請參閱服務授權參考中的 Amazon EC2 的動作、資源和條件索引鍵。

建立 Amazon VPC

複製並貼上您將使用 PCS 之 的適當 URL AWS 區域 ，以建立 VPC。您也可以下載 AWS CloudFormation 範本，並自行上傳至AWS CloudFormation 主控台。

- US East (N. Virginia) (美國東部 (維吉尼亞北部)) (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
```

- US East (Ohio) (美國東部 (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
```

- US West (Oregon) (美國西部 (奧勒岡)) (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
```

- 僅範本

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

為 PCS 建立 Amazon VPC

1. 在 AWS CloudFormation 主控台中開啟 範本。

   > **Note**
   >
   > 這些項目會預先填入範本中，以便您只保留為預設值。

2. 在提供堆疊名稱，然後堆疊名稱下，輸入 `hpc-networking`。
3. 在參數下，輸入下列詳細資訊：

   a. 在 VPC 下，然後 CidrBlock，輸入 `10.3.0.0/16`

   b. 在子網路 A 下：

      i. 然後，CidrPublicSubnetA，輸入 `10.3.0.0/20`

      ii. 然後 CidrPrivateSubnetA，輸入 `10.3.128.0/20`

   c. 在子網路 B 下：

      i. 然後，CidrPublicSubnetB，輸入 `10.3.16.0/20`

      ii. 然後 CidrPrivateSubnetA，輸入 `10.3.144.0/20`

   d. 在子網路 C 下：

      i. 針對 ProvisionSubnetsC，選取 `True`。

> **Note**
>
> 如果您要在可用區域少於三個的區域中建立 VPC，如果將 設為 ，則會忽略此選項`True`。

ii. 然後，CidrPublicSubnetB，輸入 `10.3.32.0/20`

iii. 然後 CidrPrivateSubnetA，輸入 `10.3.160.0/20`

4. 在功能下，勾選核取方塊，表示我確認 AWS CloudFormation 可能會建立 IAM 資源。

監控 AWS CloudFormation 堆疊的狀態。當 VPC 資源到達 時`CREATE_COMPLETE`，即可使用。

> **Note**
>
> 若要查看 AWS CloudFormation 範本建立的所有資源，請開啟 AWS CloudFormation 主控台。選擇 `hpc-networking` 堆疊，然後選擇 Resources (資源) 索引標籤。

AWS PCS 中的安全群組

Amazon EC2 中的安全群組充當虛擬防火牆，以控制執行個體的傳入和傳出流量。使用 AWS PCS 運算節點群組的啟動範本，將安全群組新增至其執行個體或將其移除。如果您的啟動範本不包含任何網路介面，請使用 `SecurityGroupIds`提供安全群組的清單。如果您的啟動範本定義了網路介面，您必須使用 `Groups` 參數將安全群組指派給每個網路介面。如需啟動範本的詳細資訊，請參閱 搭配 AWS PCS 使用 Amazon EC2 啟動範本。

> **Note**
>
> 對啟動範本中安全群組組態的變更只會影響運算節點群組更新後啟動的新執行個體。

安全群組需求和考量事項

AWS PCS 會在您在建立叢集時指定的子網路中建立跨帳戶彈性網路界面 (ENI)。這提供 HPC 排程器，其正在由 管理的帳戶中執行 AWS，這是與 AWS PCS 啟動的 EC2 執行個體通訊的路徑。您必須提供該 ENI 的安全群組，允許排程器 ENI 和叢集 EC2 執行個體之間的雙向通訊。

達成此目標的簡單方法是建立寬鬆的自我參考安全群組，允許群組所有成員之間所有連接埠上的 TCP/IP 流量。您可以將此附加至叢集和節點群組 EC2 執行個體。

允許的安全群組組態範例

規則類型	通訊協定	連接埠	來源	目的地
傳入	全部	全部	自我	
傳出	全部	全部		0.0.0.0/0
傳出	全部	全部		自我

這些規則允許所有流量在 Slurm 控制器和節點之間自由流動，允許所有傳出流量流向任何目的地，並啟用 EFA 流量。

限制性安全群組組態範例

您也可以限制叢集及其運算節點之間的開放連接埠。對於 Slurm 排程器，連接至叢集的安全群組必須允許下列連接埠：

- 6817 – 啟用`slurmctld`來自 EC2 執行個體的傳入連線
- 6818 – 啟用在 EC2 執行個體上執行的從 `slurmctld`到 `slurmd` 的傳出連線

連接至運算節點的安全群組必須允許下列連接埠：

- 6817 – 啟用`slurmctld`來自 EC2 執行個體的傳出連線。
- 6818 – 啟用節點群組執行個體`slurmd`上`slurmd`往返 `slurmctld`的傳入和傳出連線
- 60001–63000 – 節點群組執行個體之間的傳入和傳出連線，以支援 `srun`
- 節點群組執行個體之間的 EFA 流量。如需詳細資訊，請參閱《 Linux 執行個體使用者指南》中的準備啟用 EFA 的安全群組
- 工作負載所需的任何其他節點間流量

AWS PCS 中的多個網路介面

有些 EC2 執行個體有多個網路卡。這可讓它們提供更高的網路效能，包括超過 100 Gbps 的頻寬功能和改善的封包處理。如需具有多個網路卡之執行個體的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的彈性網路介面。

透過將網路介面新增至其 EC2 啟動範本，為 AWS PCS 運算節點群組中的執行個體設定額外的網路卡。以下是啟用兩個網路卡的範例啟動範本，例如 可在hpc7a.96xlarge執行個體上找到。請注意下列詳細資訊：

- 每個網路界面的子網路必須與您在設定將使用啟動範本的 AWS PCS 運算節點群組時選擇使用的子網路相同。
- 主要網路裝置，例如 SSH 和 HTTPS 流量等例行網路通訊，是透過設定 DeviceIndex的 來建立0。其他網路介面具有 DeviceIndex的 1。只能有一個主要網路介面，所有其他介面都是次要的。
- 所有網路介面都必須具有唯一的 NetworkCardIndex。建議的做法是按啟動範本中定義的順序對它們進行編號。
- 每個網路介面的安全群組都是使用 設定Groups。在此範例中，傳入 SSH 安全群組 (sg-*SshSecurityGroupId*) 會新增至主要網路介面，以及啟用叢集內通訊 () 的安全群組sg-*ClusterSecurityGroupId*。最後，允許對外連線至網際網路 (sg-*InternetOutboundSecurityGroupId*) 的安全群組會同時新增至主要和次要介面。

```
{
    "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"]
```

```
        }
    ]
}
```

AWS PCS 中 EC2 執行個體的置放群組

您可以使用置放群組來影響 EC2 執行個體的置放，以符合在其上執行之工作負載的需求。

置放群組類型

- 叢集 – 在可用區域中封裝緊密的執行個體，以最佳化低延遲通訊。
- 分割區 – 將執行個體分散到邏輯分割區，以協助最大化彈性。
- 分散 – 嚴格強制少數執行個體在不同的硬體上啟動，這也可以協助恢復能力。

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Amazon EC2 執行個體的置放群組。

當您將 AWS PCS 運算節點群組設定為使用 Elastic Fabric Adapter (EFA) 時，我們建議您包含叢集置放群組。

建立使用 EFA 的叢集置放群組

1. 使用運算節點群組的類型叢集建立置放群組。
   - 使用下列 AWS CLI 命令：

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

   - 您也可以使用 CloudFormation 範本來建立置放群組。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 CloudFormation 範本。從下列 URL 下載範本，並將其上傳至 CloudFormation 主控台。

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

2. 在 AWS PCS 運算節點群組的 EC2 啟動範本中包含置放群組。

搭配 AWS PCS 使用 Elastic Fabric Adapter (EFA)

Elastic Fabric Adapter (EFA) 是一種高效能的進階聯網互連 AWS ，您可以從中連接到 EC2 執行個體，以加速高效能運算 (HPC) 和機器學習應用程式。啟用在具有 EFA 的 AWS PCS 叢集上執行的應用程式需要將 AWS PCS 運算節點群組執行個體設定為使用 EFA，如下所示。

> **Note**
>
> 在與 AWS PCS 相容的 AMI 上安裝 EFA – AWS PCS 運算節點群組中使用的 AMI 必須安裝並載入 EFA 驅動程式。如需如何在安裝 EFA 軟體的情況下建置自訂 AMI 的詳細資訊，請參閱 AWS PCS 的自訂 Amazon Machine Image AMIs)。

內容

識別啟用 EFA 的 EC2 執行個體

若要使用 EFA， AWS PCS 運算群組允許的所有執行個體類型都必須支援 EFA，且必須具有相同數量 vCPUs （和 GPUs，如適用）。如需啟用 EFA 的執行個體清單，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的適用於 Amazon EC2 上 HPC 和 ML 工作負載的 Elastic Fabric Adapter。 您也可以使用 AWS CLI 來檢視支援 EFA 的執行個體類型清單。將 *region-code* 取代為您使用 AWS PCS AWS 區域 的 ，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**
>
> 判斷可用的網路介面數量 – 有些 EC2 執行個體具有多個網路卡。這可讓它們擁有多個 EFAs。如需詳細資訊，請參閱AWS PCS 中的多個網路介面。

建立安全群組以支援 EFA 通訊

AWS CLI

您可以使用下列 AWS CLI 命令來建立支援 EFA 的安全群組。命令會輸出安全群組 ID。進行下列取代：

- *region-code* – 指定您使用 AWS PCS 的 AWS 區域 ，例如 us-east-1。
- *vpc-id* – 指定您用於 AWS PCS 的 VPC ID。
- *efa-group-name* – 為安全群組提供您選擇的名稱。

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

使用下列命令來連接傳入和傳出安全群組規則。進行下列取代：

- *efa-secgroup-id* – 提供您剛建立的 EFA 安全群組 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 範本來建立支援 EFA 的安全群組。從下列 URL 下載範本，然後將其上傳至 AWS CloudFormation 主控台。

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

在 AWS CloudFormation 主控台中開啟範本時，輸入下列選項。

- 在提供堆疊名稱下
  - 在堆疊名稱下，輸入名稱，例如 `efa-sg-stack`。
- 在參數下
  - 在 SecurityGroupName 下，輸入名稱，例如 `efa-sg`。
  - 在 VPC 下，選取您將使用 AWS PCS 的 VPC。

完成建立 CloudFormation 堆疊並監控其狀態。當其到達 `CREATE_COMPLETE` EFA 安全群組時，即可使用。

（選用）建立置放群組

我們建議您啟動叢集置放群組中使用 EFA 的所有執行個體，以將兩者之間的實體距離降至最低。為您計劃使用 EFA 的每個運算節點群組建立置放群組。請參閱 AWS PCS 中 EC2 執行個體的置放群組 為您的運算節點群組建立置放群組。

建立或更新 EC2 啟動範本

EFA 網路介面是在 AWS PCS 運算節點群組的 EC2 啟動範本中設定。如果有多個網路卡，則可以設定多個 EFAs。EFA 安全群組和選用置放群組也包含在啟動範本中。

以下是具有兩個網路卡的執行個體啟動範本範例，例如 hpc7a.96xlarge。執行個體將在叢集置放群組 `subnet-`*`SubnetID1`*的 中啟動`pg-`*`PlacementGroupId1`*。

安全群組必須特別新增至每個 EFA 介面。每個 EFA 都需要啟用 EFA 流量的安全群組 (`sg-`*`EfaSecGroupId`*)。其他安全群組，尤其是處理一般流量的安全群組，例如 SSH 或 HTTPS，只需要連接到主要網路介面（由 `DeviceIndex`的 指定`0`)。定義網路介面的啟動範本不支援使用 `SecurityGroupIds` 參數設定安全群組，您必須在您設定的每個網路介面`Groups`中設定 的值。

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

建立或更新 EFA 的運算節點群組

您的 AWS PCS 運算節點群組必須包含具有相同數量 vCPUs、處理器架構和 EFA 支援的執行個體。將運算節點群組設定為搭配安裝在其上的 EFA 軟體使用 AMI，並使用設定啟用 EFA 網路介面的啟動範本。

（選用）測試 EFA

您可以執行包含在 EFA 軟體安裝中的`fi_pingpong`程式，在運算節點群組中的兩個節點之間示範已啟用 EFA 的通訊。如果此測試成功，可能是已正確設定 EFA。

若要開始，您需要在運算節點群組中執行兩個執行個體。如果您的運算節點群組使用靜態容量，則應該已有可用的執行個體。對於使用動態容量的運算節點群組，您可以使用 `salloc`命令啟動兩個節點。以下是來自叢集的範例，該叢集具有名為 的動態節點群組，該群組與名為 的佇列`hpc7g`相關聯`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 找出兩個配置節點的 IP 地址scontrol。在下列範例中，地址10.3.140.69適用於 hpc7g-1，而10.3.132.211適用於 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=24.11.5
   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=24.11.5
   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 SSH （或 SSMhpc7g-1) 連線到其中一個節點 （在此範例中為)。請注意，這是內部 IP 地址，因此如果您使用 SSH，您可能需要從其中一個登入節點進行連線。另請注意，執行個體需要透過運算節點群組啟動範本來設定 SSH 金鑰。

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

現在，fi_pingpong以伺服器模式啟動。

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

連線至第二個執行個體 (hpc7g-2)。

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

fi_pingpong 在用戶端模式下執行，連線至 上的伺服器hpc7g-1。您應該會看到類似以下範例的輸出。

```
% /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
```

（選用）使用 CloudFormation 範本建立啟用 EFA 的啟動範本

由於設定 EFA 有數個相依性，因此已提供 CloudFormation 範本，供您用來設定運算節點群組。它支援最多具有四個網路卡的執行個體。若要進一步了解具有多個網路卡的執行個體，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的彈性網路介面。

從下列 URL 下載 CloudFormation 範本，然後將其上傳至您使用 AWS PCS AWS 區域 的 中的 CloudFormation 主控台。

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

在 AWS CloudFormation 主控台中開啟範本時，輸入下列值。請注意，範本將提供一些預設參數值，您可以將它們保留為預設值。

- 在提供堆疊名稱下
  - 在堆疊名稱下，輸入描述性名稱。我們建議您合併為 AWS PCS 運算節點群組選擇的名稱，例如 *NODEGROUPNAME*-efa-lt。
- 在參數下
  - 在 NumberOfNetworkCards 下，選擇節點群組中執行個體中的網路卡數量。
  - 在 VpcId 下，選擇 AWS PCS 叢集部署所在的 VPC。
  - 在 NodeGroupSubnetId 下，選擇叢集 VPC 中要啟動啟用 EFA 執行個體的子網路。
  - 在 PlacementGroupName 下，將 欄位保留空白，以為節點群組建立新的叢集置放群組。如果您有想要使用的現有置放群組，請在此處輸入其名稱。
  - 在 ClusterSecurityGroupId 下，選擇您用來允許存取叢集中其他執行個體和 AWS PCS API 的安全群組。許多客戶會從其叢集 VPC 中選擇預設安全群組。
  - 在 SshSecurityGroupId 下，提供您用來允許傳入 SSH 存取叢集中節點的安全群組 ID。
  - 針對 SshKeyName，選取 SSH 金鑰對以存取叢集中的節點。
  - 針對 LaunchTemplateName，輸入啟動範本的描述性名稱，例如 *NODEGROUPNAME*-efa-lt。名稱對於您將使用 AWS PCS AWS 區域 的 AWS 帳戶 必須是唯一的。
- 在功能下
  - 勾選我確認 AWS CloudFormation 可能建立 IAM 資源的方塊。

監控 CloudFormation 堆疊的狀態。當它到達CREATE_COMPLETE啟動範本時，即可使用。將其與 AWS PCS 運算節點群組搭配使用，如上述 中所述建立或更新 EFA 的運算節點群組。

搭配 AWS PCS 使用網路檔案系統

您可以將網路檔案系統連接到 AWS 平行運算服務 (AWS PCS) 運算節點群組中啟動的節點，以提供資料和檔案可寫入和存取的持久性位置。您可以使用 AWS 服務提供的檔案系統，包括 Amazon Elastic File System (Amazon EFS)、Amazon FSx for Lustre、Amazon FSx for NetApp ONTAP、Amazon FSx for OpenZFS 和 Amazon File Cache。您也可以使用自我管理的檔案系統，例如 NFS 伺服器。

本主題涵蓋搭配 AWS PCS 使用網路檔案系統的考量和範例。

使用網路檔案系統的考量事項

各種檔案系統的實作詳細資訊不同，但有一些常見的考量。

- 相關檔案系統軟體必須安裝在執行個體上。例如，若要使用 Amazon FSx for Lustre，應該存在適當的Lustre套件。這可以透過在運算節點群組 AMI 中或使用執行個體開機時執行的指令碼來完成。
- 共用網路檔案系統和運算節點群組執行個體之間必須有網路路由。
- 共用網路檔案系統和運算節點群組執行個體的安全群組規則都必須允許連線至相關連接埠。
- 您必須跨存取檔案系統的資源，維持一致的POSIX使用者和群組命名空間。否則，在 PCS 叢集上執行的任務和互動式程序可能會遇到許可錯誤。
- 檔案系統掛載會使用EC2啟動範本完成。安裝網路檔案系統時發生錯誤或逾時，可能會讓執行個體無法執行任務。這反過來可能會導致意外的成本。如需有關偵錯啟動範本的詳細資訊，請參閱 搭配 AWS PCS 使用 Amazon EC2 啟動範本。

網路掛載範例

您可以使用 Amazon EFS、Amazon FSx for Lustre、Amazon FSx for NetApp ONTAP、Amazon FSx for OpenZFS 和 Amazon File Cache 建立檔案系統。展開下面的相關區段，以查看每個網路掛載的範例。

Amazon EFS

檔案系統設定

建立 Amazon EFS 檔案系統。請確定它在每個您要啟動 PCS 運算節點群組執行個體的可用區域中都有掛載目標。同時，請確定每個掛載目標都與安全群組相關聯，該安全群組允許從 PCS 運算節點群組執行個體傳入和傳出存取。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的掛載目標和安全群組。

啟動範本

將安全群組 （從您的檔案系統設定） 新增至您將用於運算節點群組的啟動範本。

包含使用cloud-config機制掛載 Amazon EFS 檔案系統的使用者資料。使用您自己的詳細資訊取代此指令碼中的下列值：

- *mount-point-directory* – 您要掛載 Amazon EFS 之每個執行個體的路徑
- *filesystem-id* – EFS 檔案系統的檔案系統 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==--
```

Amazon FSx for Lustre

檔案系統設定

在您將使用 AWS PCS 的 VPC 中建立 FSx for Lustre 檔案系統。若要將區域間傳輸降至最低，請在相同可用區域中的子網路中部署，您將在該子網路中啟動大部分的 PCS 運算節點群組執行個體。確保檔案系統與允許從 PCS 運算節點群組執行個體傳入和傳出存取的安全群組相關聯。如需安全群組的詳細資訊，請參閱《Amazon FSx for Lustre 使用者指南》中的使用 Amazon VPC 檔案系統存取控制。FSx

啟動範本

包含用於cloud-config掛載 FSx for Lustre 檔案系統的使用者資料。使用您自己的詳細資訊取代此指令碼中的下列值：

- *mount-point-directory* – 您想要掛載 FSx for Lustre 的執行個體路徑
- *filesystem-id* – FSx for Lustre 檔案系統的檔案系統 ID
- *mount-name* – FSx for Lustre 檔案系統的掛載名稱
- *region-code* – 部署 FSx for Lustre 檔案系統的 AWS 區域（必須與 AWS PCS 系統相同）
- （選用）*latest* – FSx for Lustre Lustre支援的任何版本

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

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

runcmd:
- amazon-linux-extras install -y lustre=latest
- mkdir -p /mount-point-directory
- mount -t lustre filesystem-id.fsx.region-code.amazonaws.com@tcp:/mount-name /mount-
point-directory

--==MYBOUNDARY==
```

Amazon FSx for NetApp ONTAP

檔案系統設定

在您將使用 AWS PCS 的 VPC 中建立 Amazon FSx for NetApp ONTAP 檔案系統。若要將區域間傳輸降至最低，請在相同可用區域中的子網路中部署，您將在該區域中啟動大部分的 AWS PCS 運算節點群組執行個體。請確定檔案系統與安全群組相關聯，該安全群組允許從 AWS PCS 運算節點群組執行個體傳入和傳出存取。如需安全群組的詳細資訊，請參閱 FSx for ONTAP 使用者指南中的使用 Amazon VPC 的檔案系統存取控制。

啟動範本

包含使用 `cloud-config` 來掛載 FSx for ONTAP 檔案系統根磁碟區的使用者資料。使用您自己的詳細資訊取代此指令碼中的下列值：

- *mount-point-directory* – 您想要掛載 FSx for ONTAP 磁碟區的執行個體路徑
- *svm-id* – FSx for ONTAP 檔案系統的 SVM ID
- *filesystem-id* – ONTAP 檔案系統 FSx 的檔案系統 ID

- *region-code* – 部署 FSx for ONTAP 檔案系統的 AWS 區域（必須與 AWS PCS 系統相同）
- *volume-name* – ONTAP 磁碟區的 FSx

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

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

runcmd:
- mkdir -p /mount-point-directory
- mount -t nfs svm-id.filesystem-id.fsx.region-code.amazonaws.com:/volume-name /mount-
point-directory

--==MYBOUNDARY==
```

Amazon FSx for OpenZFS

檔案系統設定

在您將使用 AWS PCS 的 VPC 中建立 FSx for OpenZFS 檔案系統。若要將區域間傳輸降至最低，請在相同可用區域中的子網路中部署，您將在該區域中啟動大部分的 AWS PCS 運算節點群組執行個體。請確定檔案系統與安全群組相關聯，該安全群組允許從 AWS PCS 運算節點群組執行個體傳入和傳出存取。如需安全群組的詳細資訊，請參閱《FSx for OpenZFS 使用者指南》中的使用 Amazon VPC 管理檔案系統存取。

啟動範本

包含使用 `cloud-config` 來掛載 FSx for OpenZFS 檔案系統根磁碟區的使用者資料。使用您自己的詳細資訊取代此指令碼中的下列值：

- *mount-point-directory* – 您想要掛載 FSx for OpenZFS 共享之執行個體的路徑
- *filesystem-id* – FSx for OpenZFS 檔案系統的檔案系統 ID
- *region-code* – 部署 AWS 區域 FSx for OpenZFS 檔案系統的（必須與 AWS PCS 系統相同）

```
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

檔案系統設定

在您將使用 AWS PCS 的 VPC 中建立 Amazon File Cache。若要將區域間傳輸降至最低，請選擇相同可用區域中的子網路，您將在該區域中啟動大部分的 PCS 運算節點群組執行個體。確保檔案快取與安全群組相關聯，該安全群組允許 PCS 執行個體和檔案快取之間連接埠 988 上的傳入和傳出流量。如需安全群組的詳細資訊，請參閱《Amazon File Cache 使用者指南》中的使用 Amazon VPC 快取存取控制。

啟動範本

將安全群組 （從您的檔案系統設定） 新增至您將用於運算節點群組的啟動範本。

包含用於`cloud-config`掛載 Amazon File Cache 的使用者資料。使用您自己的詳細資訊取代此指令碼中的下列值：

- *mount-point-directory* – 您想要掛載 FSx for Lustre 的執行個體路徑
- *cache-dns-name* – 檔案快取的網域名稱系統 (DNS) 名稱
- *mount-name* – 檔案快取的掛載名稱

```
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==
```

AWS PCS 的 Amazon Machine Image AMIs)

AWS PCS 可與您提供的 AMIs 搭配使用，為您的叢集節點上的軟體和組態提供極大的彈性。如果您嘗試使用 AWS PCS，您可以使用 提供並維護的範例 AMI AWS。如果您在生產環境中使用 AWS PCS，我們建議您建置自己的 AMIs。本主題涵蓋如何探索和使用範例 AMIs，以及如何建置和使用您自己的自訂 AMIs。

主題

搭配 AWS PCS 使用範例 Amazon Machine Image AMIs)

AWS 提供範例 AMIs，可用來做為使用 AWS PCS 的起點。

> **Important**
>
> 範例 AMIs僅供示範之用，不建議用於生產工作負載。

尋找目前的 AWS PCS 範例 AMIs

AWS Management Console

AWS PCS 範例 AMIs 具有下列命名慣例：

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

接受的值

- *####* – amzn2
- *##* – x86_64或 arm64
- *###* – slurm
- *scheduler-major-version* – 24.11

尋找 AWS PCS 範例 AMIs

1. 開啟 Amazon EC2 主控台。
2. 導覽至 AMIs。
3. 選擇公有映像。
4. 在依屬性或標籤尋找 AMI 中，使用範本名稱搜尋 AMI。

   範例

   - Arm64 執行個體上 Slurm 24.11 的範例 AMI

     ```
     aws-pcs-sample_ami-amzn2-arm64-slurm-24.11
     ```

   - 在 x86 執行個體上 Slurm 24.11 的範例 AMI

     ```
     aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11
     ```

   > **Note**
   >
   > 如果有多個 AMIs，請使用具有最新時間戳記的 AMI。

5. 當您建立或更新運算節點群組時，請使用 AMI ID。

AWS CLI

您可以使用以下命令找到最新的 AWS PCS 範例 AMI。將 *region-code* 取代為您使用 AWS PCS AWS 區域 的 ，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-24.11*' \
            '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-24.11*' \
  ```

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

當您建立或更新運算節點群組時，請使用 AMI ID。

進一步了解 AWS PCS 範例 AMIs

若要檢視 AWS PCS 範例 AMIs 目前和先前版本的內容、組態詳細資訊，請參閱 AWS PCS 範例 AMIs 版本備註。

建置與 AWS PCS 相容的自有 AMIs

若要了解如何建置可搭配 AWS PCS 使用的自有 AMIs，請參閱 AWS PCS 的自訂 Amazon Machine Image AMIs)。

AWS PCS 的自訂 Amazon Machine Image AMIs)

AWS PCS 旨在使用您帶入服務的 Amazon Machine Image (AMI)。這些 AMIs 可以安裝任意軟體和組態，只要它們已安裝 AWS PCS 代理程式和相容的 Slurm 版本並正確設定即可。您必須使用 AWS提供的安裝程式，在自訂 AMI 上安裝 AWS PCS 軟體。我們建議您使用 AWS提供的安裝程式在自訂 AMI 上安裝 Slurm，但您可以視需要自行安裝 Slurm （不建議）。

> **Note**
>
> 如果您想要在未建置自訂 AMI 的情況下嘗試 AWS PCS，您可以使用 提供的範例 AMI AWS。如需詳細資訊，請參閱搭配 AWS PCS 使用範例 Amazon Machine Image AMIs)。

本教學課程可協助您建立可與 PCS 運算節點群組搭配使用的 AMI，以支援 HPC 和 AI/ML 工作負載。

主題

- 步驟 1 – 啟動暫時執行個體
- 步驟 2 – 安裝 AWS PCS 代理程式
- 步驟 3 – 安裝 Slurm
- 步驟 4 – （選用）安裝其他驅動程式、程式庫和應用程式軟體
- 步驟 5 – 建立與 AWS PCS 相容的 AMI

步驟 1 – 啟動暫時執行個體

啟動暫時執行個體，供您用來安裝和設定 AWS PCS 軟體和 Slurm 排程器。您可以使用此執行個體來建立與 AWS PCS 相容的 AMI。

啟動暫時執行個體

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體，然後選擇啟動執行個體以開啟新的啟動執行個體精靈。
3. （選用）在名稱和標籤區段中，提供執行個體的名稱，例如 `PCS-AMI-instance`。該名稱將指派作為執行個體的資源標籤 (`Name=PCS-AMI-instance`)。
4. 在 Application and OS Images (應用程式和作業系統映像) 區段中，為其中一個支援的作業系統選取 AMI。
5. 在 Instance type (執行個體類型) 區段中，選取支援的執行個體類型。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在網路設定區段中：
    - 針對防火牆（安全群組），選擇選取現有的安全群組，然後選取允許傳入 SSH 存取執行個體的安全群組。
8. 在儲存區段中，根據需求設定磁碟區。請務必設定足夠的空間來安裝您自己的應用程式和程式庫。
9. 在 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

步驟 2 – 安裝 AWS PCS 代理程式

安裝 代理程式，設定 AWS PCS 啟動的執行個體以搭配 Slurm 使用。如需 AWS PCS 代理程式的詳細資訊，請參閱 AWS PCS 代理程式版本。

安裝 AWS PCS 代理程式

1. 連接至您啟動的執行個體。如需詳細資訊，請參閱連線至 Linux 執行個體。
2. （選用）為了確保您的所有軟體套件都是最新的，請在執行個體上執行快速軟體更新。此程序可能需要幾分鐘的時間。

- Amazon Linux 2、RHEL 9、Rocky Linux 9

```
sudo yum update -y
```

- Ubuntu 22.04

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

3. 重新啟動執行個體並重新連線至其中。
4. 下載 AWS PCS 代理程式安裝檔案。安裝檔案會封裝至壓縮的 tarball (`.tar.gz`) 檔案。若要下載最新穩定版本，請使用下列命令：使用啟動暫時執行個體 AWS 區域 的 取代*##*，例如 `us-east-1`。

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.1-1.tar.gz -o aws-pcs-agent-v1.2.1-1.tar.gz
```

您也可以在上述命令`latest`中，將版本編號取代為 ，以取得最新版本 （ 例如：`aws-pcs-agent-v1-latest.tar.gz`)。

> **Note**
>
> 這可能會在 AWS PCS 代理程式軟體的未來版本中變更。

5. （選用） 驗證 AWS PCS 軟體 tarball 的真實性和完整性。我們建議您執行這項操作來確認軟體發布者的身分，並檢查檔案自發行以來並未遭到變更或損毀。

   a. 下載 AWS PCS 的公有 GPG 金鑰，並將其匯入至 keyring。以您啟動暫時執行個體 AWS 區域 的 取代*##*。命令應傳回金鑰值。記錄金鑰值；您可以在下一個步驟中使用它。

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

   b. 執行下列命令來驗證 GPG 金鑰的指紋。

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

   命令應傳回與下列項目相同的指紋：

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

> **Important**
>
> 如果指紋不相符，請勿執行 AWS PCS 代理程式安裝指令碼。聯絡 AWS 支援。

c. 下載簽章檔案並驗證 AWS PCS 軟體 tarball 檔案的簽章。將*##*取代為您啟動暫時執行個體 AWS 區域 的 ，例如 us-east-1。

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.1-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-agent-v1.2.1-1.tar.gz.sig
```

輸出格式應類似以下內容：

```
gpg: assuming signed data in './aws-pcs-agent-v1.2.1-1.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
```

如果結果包含 ，Good signature且指紋符合上一個步驟中傳回的指紋，請繼續下一個步驟。

> **Important**
>
> 如果指紋不相符，請勿執行 AWS PCS 軟體安裝指令碼。聯絡 AWS 支援。

6. 從壓縮.tar.gz檔案擷取檔案，並導覽至擷取的目錄。

```
tar -xf aws-pcs-agent-v1.2.1-1.tar.gz && \
    cd aws-pcs-agent
```

7. 安裝 AWS PCS 軟體。

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

8. 檢查 AWS PCS 軟體版本檔案以確認安裝成功。

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

輸出格式應類似以下內容：

```
AGENT_INSTALL_DATE='Fri Dec 13 12:28:43 UTC 2024'
AGENT_VERSION='1.2.1'
AGENT_RELEASE='1'
```

步驟 3 – 安裝 Slurm

安裝與 AWS PCS 相容的 Slurm 版本。如需詳細資訊，請參閱AWS PCS 中的 Slurm 版本。

> **Note**
>
> 如果您有已安裝舊版 Slurm 軟體的 AMI，您必須執行下列步驟來安裝新版本的 Slurm。 AWS PCS 代理程式會根據叢集建立時間設定的 Slurm 版本，在執行時間啟用正確的 Slurm 二進位檔版本。

安裝 Slurm

1. 連接至安裝 AWS PCS 軟體的相同暫時執行個體。
2. 下載 Slurm 安裝程式軟體。Slurm 安裝程式封裝在壓縮的 tarball (`.tar.gz`) 檔案中。若要下載最新穩定版本，請使用下列命令：使用臨時執行個體 AWS 區域 的 取代*##*，例如 `us-east-1`。

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-
slurm-24.11-installer-24.11.5-1.tar.gz \
    -o aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz
```

您也可以在上述命令`latest`中，將版本編號取代為 ，以取得最新版本 （ 例如：`aws-pcs-slurm-24.11-installer-latest.tar.gz`)。

> **Note**
>
> 這可能會在 Slurm 安裝程式軟體的未來版本中變更。

3. （選用）驗證 Slurm 安裝程式 tarball 的真實性和完整性。我們建議您執行這項操作來確認軟體發布者的身分，並檢查檔案自發行以來並未遭到變更或損毀。

   a. 下載 AWS PCS 的公有 GPG 金鑰，並將其匯入至 keyring。使用啟動暫時執行個體 AWS 區域 的 取代*##*。命令應傳回金鑰值。記錄金鑰值；您可以在下一個步驟中使用它。

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

   b. 執行下列命令來驗證 GPG 金鑰的指紋。

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

   命令應該會傳回與下列項目相同的指紋：

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

   > **Important**
   >
   > 如果指紋不相符，請勿執行 Slurm 安裝指令碼。聯絡 AWS 支援。

   c. 下載簽章檔案並驗證 Slurm 安裝程式 tarball 檔案的簽章。將*##*取代為您啟動暫時執行個體 AWS 區域 的 ，例如 `us-east-1`。

   ```
   wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz.sig && \
       gpg --verify ./aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz.sig
   ```

   輸出格式應類似以下內容：

   ```
   gpg: assuming signed data in './aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz'
   gpg: Signature made Wed May 14 14:23:38 2025 UTC
   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
```

如果結果包含 ，Good signature且指紋符合上一個步驟中傳回的指紋，請繼續下一個步驟。

> **Important**
>
> 如果指紋不相符，請勿執行 Slurm 安裝指令碼。聯絡 AWS 支援。

4. 從壓縮的 .tar.gz 檔案中解壓縮檔案，然後導覽至解壓縮的目錄。

```
tar -xf aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz && \
    cd aws-pcs-slurm-24.11-installer
```

5. 安裝 Slurm。安裝程式會下載、編譯和安裝 Slurm 及其相依性。視您選取的暫時執行個體規格而定，這需要幾分鐘的時間。

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

6. 檢查排程器版本檔案以確認安裝。

```
cat /opt/aws/pcs/scheduler/slurm-24.11/version
```

輸出格式應類似以下內容：

```
SLURM_INSTALL_DATE='Wed May 14 14:23:38 UTC 2025'
SLURM_VERSION='24.11.5'
PCS_SLURM_RELEASE='1'
```

步驟 4 – （選用）安裝其他驅動程式、程式庫和應用程式軟體

在暫時執行個體上安裝其他驅動程式、程式庫和應用程式軟體。安裝程序會根據特定應用程式和程式庫而有所不同。如果您之前尚未為 AWS PCS 建置自訂 AMI，建議您先使用 AWS PCS 軟體和 Slurm 安裝來建置和測試 AMI，然後在確認初始成功後逐步新增自己的軟體和組態。

範例

- Elastic Fabric Adapter (EFA) 軟體。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Amazon EC2 上 HPC 工作負載的 EFA 和 MPI 入門。
- Amazon Elastic File System (Amazon EFS) 用戶端。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的手動安裝 Amazon EFS 用戶端。 Amazon Elastic File System
- Lustre 用戶端，使用 Amazon FSx for Lustre 和 Amazon File Cache。如需詳細資訊，請參閱《FSx for Lustre 使用者指南》中的安裝 Lustre 用戶端。
- Amazon CloudWatch 代理程式，以使用 CloudWatch Logs 和指標。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的安裝 CloudWatch 代理程式。 Amazon CloudWatch
- AWS Neuron，使用 trn* 和 inf* 執行個體類型。如需詳細資訊，請參閱 AWS Neuron 文件。
- NVIDIA 驅動程式、CUDA 和 DCGM，以使用 p* 或 g* 執行個體類型。

步驟 5 – 建立與 AWS PCS 相容的 AMI

安裝必要的軟體元件之後，您可以建立 AMI，重複使用該 AMI 來啟動 AWS PCS 運算節點群組中的執行個體。

從暫時執行個體建立 AMI

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您建立的暫時執行個體。選擇動作、影像、建立影像。
4. 對於 Create image (建立映像)，執行下列動作：
    a. 對於 Image name (映像名稱)，輸入 AMI 的描述性名稱。
    b. (選用) 對於 Image description (映像描述)，輸入 AMI 的簡短描述。
    c. 選擇建立映像。
5. 在導覽窗格中，選擇 AMIs。
6. 找到您在清單中建立的 AMI。等待其狀態從待定變更為可用，然後將其與 AWS PCS 運算節點群組搭配使用。

步驟 6 – 使用自訂 AMI 搭配 AWS PCS 運算節點群組

您可以將自訂 AMI 與新的或現有的 AWS PCS 運算節點群組搭配使用。

New compute node group

使用自訂 AMI

1. 開啟 AWS PCS 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要使用自訂 AMI 的叢集，然後選取運算節點群組。
4. 建立新的運算節點群組。如需詳細資訊，請參閱在 AWS PCS 中建立運算節點群組。在 AMI ID 下，搜尋您要使用的自訂 AMI 的名稱或 ID。完成設定運算節點群組，然後選擇建立運算節點群組。
5. （選用）確認 AMI 支援執行個體啟動。在運算節點群組中啟動執行個體。您可以透過將運算節點群組設定為具有單一靜態執行個體來執行此操作，也可以將任務提交至使用運算節點群組的佇列。

   a. 檢查 Amazon EC2 主控台，直到執行個體顯示以新的運算節點群組 ID 標記。如需詳細資訊，請參閱 在 AWS PCS 中尋找運算節點群組執行個體。

   b. 當您看到執行個體啟動並完成其引導程序時，請確認它正在使用預期的 AMI。若要這樣做，請選取執行個體，然後在詳細資訊下檢查 AMI ID。它應該符合您在運算節點群組設定中設定的 AMI。

   c. （選用）將運算節點群組擴展組態更新為您偏好的值。

Existing compute node group

使用自訂 AMI

1. 開啟 AWS PCS 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要使用自訂 AMI 的叢集，然後選取運算節點群組。
4. 選取您要設定的節點群組，然後選擇編輯。在 AMI ID 下，搜尋您要使用的自訂 AMI 的名稱或 ID。完成設定運算節點群組，然後選擇更新。在運算節點群組中啟動的新執行個體將使用更新的 AMI ID。現有的執行個體將繼續使用舊的 AMI，直到 AWS PCS 取代它們為止。如需詳細資訊，請參閱更新 AWS PCS 運算節點群組。
5. （選用）確認 AMI 支援執行個體啟動。在運算節點群組中啟動執行個體。您可以透過將運算節點群組設定為具有單一靜態執行個體來執行此操作，也可以將任務提交至使用運算節點群組的佇列。

a. 檢查 Amazon EC2 主控台，直到執行個體顯示以新的運算節點群組 ID 標記。如需詳細資訊，請參閱 在 AWS PCS 中尋找運算節點群組執行個體。

b. 當您看到執行個體啟動並完成其引導程序時，請確認它正在使用預期的 AMI。若要這樣做，請選取執行個體，然後在詳細資訊下檢查 AMI ID。它應該符合您在運算節點群組設定中設定的 AMI。

c. （選用）將運算節點群組擴展組態更新為您偏好的值。

步驟 7 – 終止暫時執行個體

在您確認 AMI 可如預期與 AWS PCS 搭配使用後，您可以終止暫時執行個體，以停止產生費用。

終止暫時執行個體

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您建立的暫時執行個體，然後選擇動作、執行個體狀態、終止執行個體。
4. 出現確認提示時，選擇終止。

建置 AWS PCS 自訂 AMIs 的軟體安裝程式

AWS 提供可下載的檔案，可在執行個體上安裝 AWS PCS 軟體。 AWS 也提供可下載、編譯和安裝相關版本 Slurm 及其相依性的軟體。您可以使用這些指示來建置自訂 AMIs，以便與 AWS PCS 搭配使用，也可以使用自己的方法。

內容

- AWS PCS 代理程式軟體安裝程式
- Slurm 安裝程式
- 支援的作業系統
- 支援的執行個體類型
- 支援的 Slurm 版本
- 使用檢查總和驗證安裝程式

AWS PCS 代理程式軟體安裝程式

AWS PCS 代理程式軟體安裝程式會將執行個體設定為在執行個體引導程序期間使用 AWS PCS。您必須使用 AWS提供的安裝程式，在自訂 AMI 上安裝 AWS PCS 代理程式。

如需 AWS PCS 代理程式軟體的詳細資訊，請參閱 AWS PCS 代理程式版本。

Slurm 安裝程式

Slurm 安裝程式會下載、編譯和安裝 Slurm 及其相依性的相關版本。您可以使用 Slurm 安裝程式來建置 AWS PCS AMIs。如果自己的機制與 Slurm 安裝程式提供的軟體組態一致，您也可以使用自己的機制。如需 Slurm 的 AWS PCS 支援詳細資訊，請參閱 AWS PCS 中的 Slurm 版本。

AWS提供的軟體會安裝下列項目：

- 請求的主要和維護版本（目前為 24.11.x 版）上的 Slurm - 授權 GPL 2
  - Slurm 建置時將 `--sysconfdir` 設為 `/etc/slurm`
  - Slurm 使用 選項`--enable-pam`和 建置 `--without-munge`
  - Slurm 是使用 選項建置 `--sharedstatedir=/run/slurm/`
  - Slurm 使用 PMIX 和 JWT 支援建置
  - Slurm 安裝在 `/opt/aws/pcs/schedulers/slurm-24.11`
- OpenPMIX (4.2.6 版） – 授權
  - OpenPMIX 已安裝為 的子目錄 `/opt/aws/pcs/scheduler/`
- libjwt (1.17.0 版） – 授權 MPL-2.0
  - libjwt 已安裝為 的子目錄 `/opt/aws/pcs/scheduler/`

AWS提供的軟體會變更系統組態，如下所示：

- 組建建立的 Slurm `systemd` 檔案會複製到檔案名稱為 `/etc/systemd/system/`的 `slurmd-24.11.service`。
- 如果不存在，則會使用 UID/GID 建立 Slurm 使用者和群組 (`slurm:slurm`)`401`。
- 在 Amazon Linux 2 和 Rocky Linux 9 上，安裝會新增 EPEL 儲存庫，以安裝建置 Slurm 或其相依性所需的軟體。
- 在 RHEL9 上， 安裝將啟用 `fedoraproject codeready-builder-for-rhel-9-rhui-rpms`和 `epel-release-latest-9` 從 安裝必要的軟體，以建置 Slurm 或其相依性。

支援的作業系統

請參閱 AWS PCS 中支援的作業系統。

> **Note**
>
> AWS 深度學習 AMIs 以 Amazon Linux 2 和 Ubuntu 22.04 為基礎的 (DLAMI) 版本應與 AWS PCS 軟體和 Slurm 安裝程式相容。如需詳細資訊，請參閱《 AWS 深度學習 AMIs 開發人員指南》中的選擇您的 DLAMI。

支援的執行個體類型

AWS PCS 軟體和 Slurm 安裝程式支援任何 x86_64 或 arm64 執行個體類型，而 可執行其中一個支援的作業系統。

支援的 Slurm 版本

請參閱 AWS PCS 中的 Slurm 版本。

使用檢查總和驗證安裝程式

您可以使用 SHA256 檢查總和來驗證安裝程式 tarball (.tar.gz) 檔案。我們建議您執行這項操作來確認軟體發行者的身分識別，並檢查應用程式自發行以來並未遭到變更或損毀。

驗證 tarball

針對 SHA256 檢查總和使用 sha256sum 公用程式，並指定 tarball 檔案名稱。 SHA256 您必須從儲存 tarball 檔案的目錄中執行 命令。

- SHA256

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

命令應以下列格式傳回檢查總和值。

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

比較 命令傳回的檢查總和值與下表中提供的檢查總和值。如果檢查總和相符，則可以安全地執行安裝指令碼。

> **Important**
>
> 如果檢查總和不相符，請勿執行安裝指令碼。請聯絡 支援。

例如，下列命令會產生 Slurm 24.11.5-1 tarball 的 SHA256 檢查總和。

```
$ sha256sum aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz
```

輸出範例：

```
593efe4d66bef2f3e46d5a382fb5a32f7a3ca2510bcf1b3c85739f4f951810d5 aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz
```

下表列出安裝程式最新版本的檢查總和。將 *us-east-1* 取代為您使用 AWS PCS AWS 區域 的 。

AWS PCS 代理程式

Installer (安裝程式)	下載 URL	SHA256 檢查總和
AWS PCS 代理程式 1.2.1-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.1-1.tar.gz`	`2b784643ca01ccca1baa64fbfb34bb41efe8bdca69470998b74ce3962bc271d4`
AWS PCS 代理程式 1.2.0-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.0-1.tar.gz`	`470db8c4fc9e50277b6317f98584b6b547e73523043e34f018eecae767846805`
AWS PCS 代理程式 1.1.1-1	`https://aws-pcs-repo- us-east-1 .s3.us-`	`bef078bf60a6d8ecde2e6c49cd34d088703f`

Installer (安裝程式)	下載 URL	SHA256 檢查總和
	east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz	02550279e3bf483d57a235334dc6
AWS PCS 代理程式 1.1.0-1	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.0-1.tar.gz	594c32194c71bccc5d66e5213213ae38dd2c6d2f9a950bb01accea0bbab0873a
AWS PCS 代理程式 1.0.1-1	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.1-1.tar.gz	04e22264019837e3f42d8346daf5886eaacecd21571742eb505ea8911786bcb2
AWS PCS 代理程式 1.0.0-1	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz	d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042144b134ce0

Slurm 安裝程式

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 24.11.5-1	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz	593efe4d66bef2f3e46d5a382fb5a32f7a3ca2510bcf1b3c85739f4f951810d5

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 24.05.7-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.7-1.tar.gz`	`0b5ed7c81195de2628c78f37c79e63fc4ae99132ca6b019b53a0d68792ee82c5`
Slurm 24.05.5-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz`	`7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b`
Slurm 23.11.10-3	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-3.tar.gz`	`488a10ee0fbd57ec0e0ff7ea708a9e3038fafdc025c6bb391c75c2e2a7852a00`
Slurm 23.11.10-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-2.tar.gz`	`0bbe85423305c05987931168caf98da08a34c25f9eec0690e8e74de0b7bc8752`

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 23.11.10-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-1.tar.gz`	`27e8faa9980e92cdfd8cfdc71f937777f0934552ce61e33dac4ecf5a20321e44`
Slurm 23.11.9-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz`	`1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8`

AWS PCS 範例 AMIs版本備註

最新支援的排程器主要版本的 AMIs 會收到安全性更新和重大錯誤修正。這些增量安全修補程式不包含在官方版本備註中。

> **Important**
>
> 不支援與舊排程器版本相關的範例 AMIs，也不會接收更新。

> **Important**
>
> 範例 AMIs僅供示範之用，不建議用於生產工作負載。

內容

AWS x86_64 的 PCS 範例 AMIs (Amazon Linux 2)

Slurm 24.11

> **Note**
>
> AWS PCS 支援 Slurm 24.11 和更新版本。如需詳細資訊，請參閱AWS PCS 中的低語會計。

AMI 名稱

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11`

支援的 EC2 執行個體

- 具有 64 位元 x86 處理器的所有執行個體。若要尋找相容的執行個體，請導覽至 Amazon EC2 主控台。選擇執行個體類型，然後搜尋 `Architectures=x86_64`。

AMI 內容

- 支援 AWS 的服務： AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構：x86_64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.33.0
- GDRCopy：2.4
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

Slurm 24.05

AMI 名稱

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

支援的 EC2 執行個體

- 具有 64 位元 x86 處理器的所有執行個體。若要尋找相容的執行個體，請導覽至 Amazon EC2 主控台。選擇執行個體類型，然後搜尋 `Architectures=x86_64`。

AMI 內容

- 支援 AWS 的服務： AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構：x86_64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.33.0
- GDRCopy：2.4
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

Slurm 23.11

AMI 名稱

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

支援的 EC2 執行個體

- 具有 64 位元 x86 處理器的所有執行個體。若要尋找相容的執行個體，請導覽至 Amazon EC2 主控台。選擇執行個體類型，然後搜尋 `Architectures=x86_64`。

AMI 內容

- 支援 AWS 的服務： AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構：x86_64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.33.0
- GDRCopy：2.4

- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

AWS Arm64 (Amazon Linux 2) 的 PCS 範例 AMIs

Slurm 24.11

> **Note**
>
> AWS PCS 支援 Slurm 24.11 和更新版本。如需詳細資訊，請參閱AWS PCS 中的低語會計。

AMI 名稱

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.11`

支援的 EC2 執行個體

- 具有 64 位元 Arm 處理器的所有執行個體。若要尋找相容的執行個體，請導覽至 Amazon EC2 主控台。選擇執行個體類型，然後搜尋 `Architectures=arm64`。

AMI 內容

- 支援 AWS 的服務： AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構： arm64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.33.0
- GDRCopy：2.4
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

Slurm 24.05

AMI 名稱

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

支援的 EC2 執行個體

- 具有 64 位元 Arm 處理器的所有執行個體。若要尋找相容的執行個體，請導覽至 Amazon EC2 主控台。選擇執行個體類型，然後搜尋 `Architectures=arm64`。

AMI 內容

- 支援 AWS 的服務： AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構： arm64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.33.0
- GDRCopy：2.4
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

Slurm 23.11

AMI 名稱

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

支援的 EC2 執行個體

- 具有 64 位元 Arm 處理器的所有執行個體。若要尋找相容的執行個體，請導覽至 Amazon EC2 主控台。選擇執行個體類型，然後搜尋 `Architectures=arm64`。

AMI 內容

- 支援 AWS 的服務： AWS PCS

- 作業系統：Amazon Linux 2
- 運算架構： arm64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.33.0
- GDRCopy：2.4
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

AWS PCS 中支援的作業系統

AWS PCS 使用為運算節點群組設定的 Amazon Machine Image (AMI)，在該運算節點群組中啟動 EC2 執行個體。AMI 會決定 EC2 執行個體使用的作業系統。您無法變更 AWS PCS 範例 AMIs中的作業系統。如果您想要使用不同的作業系統，則必須建立自訂 AMI。如需詳細資訊，請參閱AWS PCS 的 Amazon Machine Image AMIs)。

支援的作業系統

- Amazon Linux 2

  這是 AWS PCS 範例 AMIs中的作業系統。

  > **Important**
  >
  > 範例 AMIs僅供示範之用，不建議用於生產工作負載。您應該為生產工作負載建立並使用自訂 AMI，即使您打算使用 Amazon Linux 2。

- RedHat Enterprise Linux 9 (RHEL 9)

  RHEL 任何執行個體類型的隨需成本高於其他支援的作業系統。如需定價的詳細資訊，請參閱隨需定價和 Amazon Elastic Compute Cloud 上的 Red Hat Enterprise Linux 如何提供和定價？。

- Rocky Linux 9

  您可以使用官方 Rocky Linux 9 AMIs 做為自訂 AMI 的基礎。如果基本 AMI 沒有最新的核心，您的自訂 AMI 建置可能會失敗。

  升級核心

  1. 從這裡使用 rocky9 AMI ID 啟動執行個體：https：//https://rockylinux.org/cloud-images/
  2. ssh 至執行個體並執行下列命令：

     ```
     sudo yum -y update
     ```

  3. 從執行個體建立映像。您可以將此映像指定為自訂 AMI `ParentImage`的 。

- Ubuntu 22.04

  Ubuntu 22.04 需要更安全的 SSH 金鑰，預設不支援 RSA 金鑰。我們建議您產生並使用 ED25519 金鑰。

AWS PCS 代理程式版本

AWS PCS 代理程式軟體會設定 EC2 執行個體 AWS PCS 啟動，以便與 Slurm 搭配使用。您在為叢集建立運算節點群組時指定的 Amazon Machine Image (AMI) 中包含代理程式。在這些運算節點群組中啟動的 EC2 執行個體會使用指定的 AMI 及其隨附的 AWS PCS 代理程式軟體。 AWS PCS 代理程式可讓 EC2 執行個體將自己註冊為叢集的一部分。若要使用最新的 AWS PCS 代理程式軟體，您必須更新自訂 AMIs。如需詳細資訊，請參閱 AWS PCS 的自訂 Amazon Machine Image AMIs) 中的 步驟 2 – 安裝 AWS PCS 代理程式。

AWS PCS 代理程式版本	版本日期	版本備註
1.2.0-1 版	2025 年 3 月 7 日	• 在 中啟用 IPv6 支援`slurmd.conf` 。
1.1.1-1 版	2024 年 12 月 13 日	• 修正呼叫 RegisterComputeNodeGroupInstance 時回報不正確 Slurm 版本的問題。 • 修正在 中`/opt/aws/pcs/etc/bootstrap_hooks/` 執行自訂指令碼時，執行個體中繼資料未正確擷取的問題。
1.1.0-1 版	2024 年 12 月 6 日	• 在 中啟用自訂指令碼`/opt/aws/pcs/etc/bootstrap_hooks/` ，以在引導步驟之前執行。
1.0.1-1 版	2024 年 10 月 22 日	• 修正 NVIDIA 裝置在啟用 GPU 的執行個體上`slurmd`啟動時無法運作的問題。
1.0.0-1 版	2024 年 8 月 28 日	• 初始版本。

AWS PCS 中的 Slurm 版本

SchedMD 透過新功能、最佳化和安全性修補程式持續增強 Slurm。SchedMD 會定期發行新的主要版本，並計劃在任何指定時間支援最多 3 個版本。 AWS PCS 旨在使用修補程式版本自動更新 Slurm 控制器。

當 SchedMD 結束對特定主要版本的支援時， AWS PCS 也會結束對該主要版本的支援。如果 Slurm 主要版本接近生命週期結束， AWS PCS 會傳送預先通知，協助客戶知道何時將其叢集升級到較新的支援版本。

我們建議您使用最新支援的 Slurm 版本來部署叢集，以存取最新的進展和改進。

AWS PCS 中支援的 Slurm 版本

下表顯示支援的 Slurm 版本，以及每個版本的重要日期和資訊。

Slurm 版本	SchedMD 發行日期	AWS PCS 發行日期	AWS PCS 支援結束日期	最低相容 AWS PCS 代理程式版本	支援的 AWS PCS 範例 AMIs
24.11	11/29/2024	5/14/2025	5/31/2026	1.0.0-1	• aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11 • aws-pcs-sample_ami-amzn2-arm64-slurm-24.11
24.05	5/30/2024	12/18/2024	11/30/2025	1.0.0-1	• aws-pcs-s

Slurm 版本	SchedMD 發行日期	AWS PCS 發行日期	AWS PCS 支援結束日期	最低相容 AWS PCS 代理程式版本	支援的 AWS PCS 範例 AMIs
					ample_ami-amzn2-x86_64-slurm-24.05 • aws-pcs-sample_ami-amzn2-arm64-slurm-24.05

AWS PCS 中不支援的 Slurm 版本

下表顯示 AWS PCS 中不支援的 Slurm 版本。

Slurm 版本	SchedMD 發行日期	AWS PCS 發行日期	AWS PCS 支援結束日期		
23.11	11/21/2023	8/28/2024	5/31/2025		

AWS PCS 中 Slurm 版本的版本備註

本主題說明 AWS PCS 目前支援的每個 Slurm 版本的重要變更。建議您在升級叢集時，檢閱新舊版本之間的變更。

Slurm 24.11

在 AWS PCS 中實作的變更

- AWS PCS 支援 Slurm 會計。如需詳細資訊，請參閱AWS PCS 中的低語會計。

如需 Slurm 24.11 的詳細資訊，請參閱下列出版物：

- SchedMD 版本公告
- SchedMD 版本備註

Slurm 24.05

在 AWS PCS 中實作的變更

- 新的 Slurm Step Manager 模組現在預設為在 AWS PCS 中啟用。本單元透過將步驟管理從中央控制器卸載到運算節點，大幅改善具有大量步驟用量的環境中的系統並行性，提供顯著的好處。為了支援此組態和更好的隔離`Prolog`和`Epilog`程序執行，會啟用新的 prolog 旗標 (`Contain`、`Alloc`)。
- 啟用從控制器到運算節點的階層式通訊，以最佳化 Slurm 節點內通訊，進而改善可擴展性和效能。此外，路由組態現在使用分割區節點清單從控制器進行通訊，而不是外掛程式的預設路由演算法，以增強系統彈性。
- 新的雜湊外掛程式會`HashPlugin=hash/sha3`取代先前的 `hash/k12 plugin`。這現在預設為在 AWS PCS 叢集中啟用。
- Slurm 控制器日誌現在包含對所有傳入遠端程序呼叫 (RPC) 的增強型稽核功能`slurmctld`。日誌包含來源地址、已驗證的使用者，以及連線處理之前的 RPC 類型。

如需 Slurm 24.05 的詳細資訊，請參閱下列出版物：

- SchedMD 版本公告
- SchedMD 版本備註

Slurm 23.11

您可以在 AWS PCS 中變更 Slurm 設定

- `SuspendTime` 預設為 `60`。使用 AWS PCS `scaleDownIdleTimeInSeconds`組態參數進行設定。如需詳細資訊，請參閱 AWS PCS API 參考中 `ClusterSlurmConfiguration` 資料類型的 `scaleDownIdleTimeInSeconds` 參數。
- `MaxJobCount` 和 `MaxArraySize`是根據您為叢集選擇的大小。如需詳細資訊，請參閱 PCS `CreateCluster` API 參考中 API 動作的 `size` 參數。 AWS

- `SelectTypeParameters` Slurm 設定預設為 `CR_CPU`。您可以提供它做為值`slurmCustomSettings`，讓 在建立叢集時設定它。如需詳細資訊，請參閱 AWS PCS API 參考中 `CreateCluster` API 動作和 SlurmCustomSetting 的 `slurmCustomSettings` 參數。
- 您可以在叢集層級設定 `Epilog` `Prolog`和 。您可以提供它做為值`slurmCustomSettings`，讓 在建立叢集時設定它。如需詳細資訊，請參閱 AWS PCS API 參考中的 `CreateCluster`和 SlurmCustomSetting。
- 您可以在運算節點群組層級設定 `RealMemory` `Weight`和 。您可以在建立運算節點群組時，提供它做為 的值`slurmCustomSettings`。如需詳細資訊，請參閱 AWS PCS API 參考中的 `CreateComputeNodeGroup`和 SlurmCustomSetting。

有關 AWS PCS 中 Slurm 版本的常見問題

AWS PCS 維持對多個 Slurm 版本的支援。推出新的 Slurm 版本時， AWS PCS 會提供技術支援和安全修補程式，直到該版本達到 SchedMD. AWS PCS 的終止支援 (EOS) 為止，參考 Slurm 版本的 EOS 日期作為生命週期結束 (EOL)，以符合 AWS 術語。

AWS PCS 支援 Slurm 版本多久？

AWS Slurm 版本的 PCS 支援符合 SchedMD 對主要版本的支援週期。 AWS PCS 支援目前版本和 2 個最新的先前主要版本。當 SchedMD 發行新的主要版本時， AWS PCS 會終止對最舊支援版本的支援。 AWS PCS 會盡快發行 Slurm 的新主要版本，但 SchedMD 的發行與其在 AWS PCS 中的可用性之間可能會有延遲。

我的叢集如何取得新的 Slurm 修補程式版本版本？

為了解決錯誤和安全性修正， AWS PCS 旨在自動將修補程式套用至在內部服務擁有帳戶中執行的叢集控制器。若要在 的 EC2 執行個體上安裝修補程式 AWS 帳戶，請更新運算節點群組的 Amazon Machine Image (AMI)，並更新運算節點群組以使用更新的 AMI。如需詳細資訊，請參閱AWS PCS 的自訂 Amazon Machine Image AMIs)。

> **Note**
>
> 更新時無法使用 Slurm 控制器。執行中的任務不會受到影響。在叢集的控制器無法使用之前提交的任務會保留，直到控制器可用為止。

如何通知即將發生的 Slurm 版本 EOL 事件？

我們會在 EOL 日期前 6 個月傳送電子郵件訊息給您。我們每個月都會在 EOL 之前傳送電子郵件訊息給您，並在 EOL 日期前 1 週收到最終電子郵件訊息。在 EOL 日期之後，我們會每月傳送電子郵件訊息 12 個月給使用 EOL Slurm 版本執行 AWS PCS 叢集的客戶。如果該版本已識別安全漏洞，我們可能會暫停具有 EOL Slurm 版本的叢集。

如何判斷叢集使用的 Slurm 版本是否正在執行 EOL Slurm 版本？

我們會傳送電子郵件訊息給您，通知您有一個執行中叢集具有 EOL Slurm 版本。我們會發佈提醒，AWS Health Dashboard 其中包含具有 EOL Slurm 版本的叢集詳細資訊。您也可以使用 AWS PCS 主控台來識別具有 EOL Slurm 版本的叢集。

如果我的 Slurm 版本接近或超過 EOL，我需要做什麼？

使用較新支援的 Slurm 版本建立新的叢集，並在運算節點群組 AMIs 中更新 Slurm 版本。AMIs和執行中 EC2 執行個體中的 Slurm 版本不能超過叢集 Slurm 版本的 2 個版本。如需詳細資訊，請參閱AWS PCS 的自訂 Amazon Machine Image AMIs)。

如果我未在 EOL 日期之前切換到較新版本的 Slurm，會發生什麼情況？

您無法使用 EOL Slurm 版本建立新的叢集。現有的叢集可以在沒有 AWS 支援的情況下運作長達 12 個月，而且不需要立即採取動作來維護其操作。在 EOL 日期之後，無法保證支援、安全性更新和可用性。基於安全考量，我們可能會暫停叢集。我們強烈建議您使用支援的 Slurm 版本來維護 AWS PCS 叢集的安全性和支援。

使用 EOL Slurm 版本操作叢集有哪些風險？

具有 EOL Slurm 版本的叢集具有顯著的安全性和操作風險。如果沒有 SchedMD 的主動監控，安全漏洞可能會保持未偵測到或未解決的狀態。如果發現重大漏洞，我們可能會立即暫停您的叢集。

當我的叢集暫停時，我的任務、叢集運算、儲存和聯網資源會發生什麼情況？

AWS PCS 管理的所有資源都會終止。這包括 Slurm 控制器、運算節點群組和 EC2 執行個體。在運算執行個體上執行的任何任務都會立即終止，且叢集會進入暫停狀態。客戶管理的資源，例如外部檔案系統，保持不變。您可以使用 AWS PCS 主控台和 API 動作來存取叢集的組態。

我可以重新啟動暫停的叢集以繼續其剩餘的任務嗎？

否，您無法重新啟動暫停的叢集。您可以使用暫停叢集的組態，建立具有支援 Slurm 版本的新叢集。如果您將剩餘的任務儲存在外部檔案系統中，則可以執行這些任務。

我可以請求超過 12 個月寬限期的延長嗎？

否，您無法請求擴充功能來執行超過 12 個月寬限期的叢集。我們提供較長的時間，協助您切換到支援的 Slurm 版本。為了避免中斷叢集操作，建議您在 Slurm 版本達到 EOL 之前進行切換。

AWS PCS 中的低語會計

您可以啟用新 AWS PCS 叢集的會計，以監控叢集用量、強制執行資源限制，以及管理特定佇列或運算節點群組的精細存取控制。 AWS PCS 會建立和管理叢集的會計資料庫，讓您無需建立和管理自己的個別會計資料庫。 AWS PCS 會使用 Slurm 中的會計功能。如需 Slurm 中會計功能的詳細資訊，請參閱 SchedMD 中的 Slurm 文件。

若要使用會計，請在建立新叢集時啟用它，並選擇性地設定會計參數。叢集狀態為 `Active`且具有運算節點群組後，您可以連線至登入節點的 Linux shell 來執行會計函數，例如使用 Slurm `sacct`命令檢視任務資料。

> **Note**
>
> Slurm 24.11 或更新版本支援會計。

AWS PCS console

在建立叢集頁面上，您必須選取有效的 Slurm 版本 (24.11 版或更新版本）。在排程器設定下，啟用計費。

AWS PCS API

在您呼叫 `CreateCluster` API 動作時提供`accounting`組態。在 `accounting` 物件中，將 `mode`設定為 `STANDARD`。如需詳細資訊，請參閱 AWS PCS API 參考中的 CreateCluster 和會計。

下列範例使用 AWS CLI 呼叫 `CreateCluster` API 動作。參數值子字串`accounting='{mode=STANDARD}'`會啟用會計。

```
aws pcs create-cluster --cluster-name cluster-name \
                       --scheduler type=SLURM,version=24.11 \
                       --size SMALL \
                       --networking subnetIds=cluster-subnet-
id,securityGroupIds=cluster-security-group-id \
                       --slurm-configuration
 scaleDownIdleTimeInSeconds=180,accounting='{mode=STANDARD}',slurmCustomSettings='[{paramete
```

> **Important**
>
> 如果您啟用 會計，則會收取額外的帳單費用。如需詳細資訊，請參閱 AWS PCS 定價頁面。

> **Important**
>
> 您無法在已啟用它的叢集上停用會計。您必須刪除叢集。

AWS PCS 中 Slurm 會計的重要概念

下列概念專屬於 AWS PCS，並控制 AWS PCS 如何實作 Slurm 會計。

會計資料庫

AWS PCS 會將您的會計資料存放在在 AWS 擁有的 中建立 AWS 帳戶 的資料庫中。您無法存取 `slurmdbd.conf`.

預設清除時間

此 AWS PCS 設定會指定所有會計記錄類型的保留期間 （天數）（任務、事件、保留、步驟、暫停、交易、用量資料）。例如，如果值為 30， AWS PCS 會保留會計記錄 30 天。您在建立叢集時提供此值。如果您不提供值， AWS PCS 會無限期保留資料庫中的會計記錄。

AWS PCS console

您可以在建立叢集的步驟中指定預設清除時間。在建立叢集頁面上，您必須選取有效的 Slurm 版本 (24.11 版或更新版本） 並啟用會計。在排程器設定下，提供預設清除時間 （天） 的整數值。

AWS PCS API

指定 `defaultPurgeTimeInDays`做為`accounting`您在 API `CreateCluster`動作呼叫中提供的資訊的一部分。如需詳細資訊，請參閱 AWS PCS API 參考中的 CreateCluster 和會計。

> **Note**
>
> 當您使用 AWS PCS API 建立叢集時， 的預設值`defaultPurgeTimeInDays`為 `-1`，且 `0` 不是有效的值。

會計政策強制執行

此設定會決定 Slurm 如何嚴格為您的叢集強制執行任務提交規則、資源限制和會計政策。此設定對應至叢集`slurm.conf`檔案中的 `AccountingStorageEnforce` 參數。您可以選取強制執行選項的任意組合。如果您未選取任何選項，則叢集上的任務不會套用任何會計限制。 AWS PCS 支援下列選項：

- 關聯 — job-to-account映射
- 限制 — 資源限制
- QoS — 服務品質要求
- 安全模式 — 保證在限制內完成
- nosteps — 停用步驟會計
- nojobs — 停用任務會計

如需這些選項的詳細資訊，請參閱 SchedMD 中的 Slurm 文件。

AWS PCS console

您可以在建立叢集的步驟中設定選項。在建立叢集頁面上，您必須選取有效的 Slurm 版本 (24.11 版或更新版本） 並啟用會計。從排程器設定下的會計政策強制執行下拉式清單中選取您想要的選項。

AWS PCS API

在 Slurm 中，這些選項是在叢集的 `slurm.conf` 檔案中設定。您無法直接存取 AWS PCS 叢集`slurm.conf`的 。反之，您可以在建立叢集時`SlurmCustomSettings`將 提供給 `CreateCluster` API 動作。如需詳細資訊，請參閱 AWS PCS API 參考中的 CreateCluster。

取得現有 AWS PCS 叢集的會計組態

Slurm 會計組態包含在叢集的 Slurm 組態中。

AWS PCS console

1. 從導覽窗格中選擇叢集。
2. 從清單中選擇叢集名稱。
3. 在組態索引標籤上，尋找 Slurm 組態下的會計組態

AWS PCS API

使用 `GetCluster` API 動作來取得叢集組態。您可以在 中找到會計組態`slurmConfiguration`。的設定`mode`和 的值`defaultPurgeTimeInDays`低於 `accounting`。選取的會計政策強制執行選項位於 下`slurmCustomSettings`。如需詳細資訊，請參閱 AWS PCS API 參考中的 GetCluster。

AWS 平行運算服務的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了符合最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。作為AWS 合規計畫的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 AWS 平行運算服務的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 AWS PCS 時套用共同責任模型。下列主題說明如何設定 AWS PCS 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS PCS 資源。

主題

- AWS 平行運算服務中的資料保護
- AWS Parallel Computing 服務 使用界面端點存取 (AWS PrivateLink)
- AWS 平行運算服務的 Identity and Access Management
- AWS 平行運算服務的合規驗證
- AWS 平行運算服務的彈性
- AWS 平行運算服務中的基礎設施安全性
- Parallel Computing Service AWS 中的漏洞分析和管理
- 預防跨服務混淆代理人
- AWS Parallel Computing Service 的安全最佳實務

AWS 平行運算服務中的資料保護

AWS 共同責任模型適用於 AWS 平行運算服務中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需

有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 AWS PCS 或使用主控台、API AWS CLI或 AWS SDKs的其他 AWS 服務 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

靜態加密

當您使用 AWS Management Console、、AWS PCS API 或 AWS SDKs 建立 AWS 平行運算服務 (AWS CLI AWS PCS) 叢集時，預設會針對靜態資料啟用加密。 AWS PCS 使用 AWS 擁有的 KMS 金鑰來加密靜態資料。如需詳細資訊，請參閱《 AWS KMS 開發人員指南》中的客戶金鑰和 AWS 金鑰。您也可以使用客戶受管金鑰。如需詳細資訊，請參閱在 AWS PCS 中與加密 EBS 磁碟區搭配使用所需的 KMS 金鑰政策。

叢集秘密存放在 中， AWS Secrets Manager 並使用 Secrets Manager 受管 KMS 金鑰加密。如需詳細資訊，請參閱在 AWS PCS 中使用叢集秘密。

在 AWS PCS 叢集中，下列資料為靜態：

- 排程器狀態 – 它包含叢集中執行中任務和佈建節點的資料。這是 Slurm 在中`StateSaveLocation`定義的 中保留的資料`slurm.conf`。如需詳細資訊，請參閱 Slurm 文件中的 StateSaveLocation 說明。 AWS PCS 會在任務完成後刪除任務資料。

- 排程器身分驗證秘密 – AWS PCS 會使用它來驗證叢集中的所有排程器通訊。

對於排程器狀態資訊， AWS PCS 會在將資料和中繼資料寫入檔案系統之前自動加密資料和中繼資料。加密的檔案系統針對靜態資料使用業界標準的 AES-256 加密演算法。

傳輸中加密

無論您是否使用 AWS Command Line Interface (AWS CLI) 或 AWS SDKs， AWS PCS API 的連線都會搭配 Signature 第 4 版簽署程序使用 TLS 加密。如需詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的簽署 AWS API 請求。透過 API AWS 管理存取控制，其中包含您用於連線之安全登入資料的 IAM 政策。

AWS PCS 使用 TLS 連線到其他 AWS 服務。

在 Slurm 叢集中，排程器使用身分`auth/slurm`驗證外掛程式設定，該外掛程式為所有排程器通訊提供身分驗證。Slurm 不會在應用程式層級為其通訊提供加密，在叢集執行個體間流動的所有資料都會保留在 EC2 VPC 本機，因此如果這些執行個體支援傳輸中的加密，則受 VPC 加密約束。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的傳輸中加密。控制器 （ 在服務帳戶中佈建 ） 與您帳戶中叢集節點之間的通訊會加密。

金鑰管理

AWS PCS 使用 AWS 擁有的 KMS 金鑰來加密資料。如需詳細資訊，請參閱《 AWS KMS 開發人員指南》中的客戶金鑰和 AWS 金鑰。您也可以使用客戶受管金鑰。如需詳細資訊，請參閱在 AWS PCS 中與加密 EBS 磁碟區搭配使用所需的 KMS 金鑰政策。

叢集秘密存放在 中， AWS Secrets Manager 並使用 Secrets Manager 受管 KMS 金鑰加密。如需詳細資訊，請參閱在 AWS PCS 中使用叢集秘密。

網際網路流量隱私權

AWS 叢集的 PCS 運算資源位於客戶帳戶中的 1 個 VPC 內。因此，叢集內的所有內部 AWS PCS 服務流量都會保留在 AWS 網路中，而不會周遊網際網路。使用者和 AWS PCS 節點之間的通訊可以通過網際網路，我們建議您使用 SSH 或 Systems Manager 連接到節點。如需詳細資訊，請參閱AWS Systems Manager 《 使用者指南》中的什麼是 AWS Systems Manager？。

您也可以使用下列方案，將內部部署網路連線至 AWS：

- AWS Site-to-Site VPN。 如需詳細資訊，請參閱AWS Site-to-Site VPN 《 使用者指南》中的什麼是 AWS Site-to-Site VPN？。

- AWS Direct Connect。如需詳細資訊，請參閱AWS Direct Connect 《 使用者指南》中的什麼是 AWS Direct Connect？。

您可以存取 AWS PCS API 來執行服務的管理任務。您和您的使用者會存取 Slurm 端點連接埠，以直接與排程器互動。

加密 API 流量

若要存取 AWS PCS API，用戶端必須支援 Transport Layer Security (TLS) 1.2 或更新版本。我們需要 TLS 1.2 並建議使用 TLS 1.3。用戶端也必須支援具備完整轉寄密碼 (PFS) 的密碼套件，例如暫時性 Diffie-Hellman (DHE) 或橢圓曲線 Diffie-Hellman Ephemeral (ECDHE)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。您也可以使用 AWS Security Token Service (AWS STS) 產生臨時安全登入資料來簽署請求。

加密資料流量

從存取排程器端點的支援 EC2 執行個體，以及從 內的 ComputeNodeGroup 執行個體之間啟用傳輸中的資料加密 AWS 雲端。如需詳細資訊，請參閱傳輸中加密。

在 AWS PCS 中與加密 EBS 磁碟區搭配使用所需的 KMS 金鑰政策

AWS PCS 使用服務連結角色將許可委派給其他 AWS 服務。 AWS PCS 服務連結角色已預先定義，並包含 AWS PCS AWS 服務 代表您呼叫其他 所需的許可。預先定義的許可也包括存取您的 AWS 受管金鑰 ，但無法存取您的客戶受管金鑰。

本主題說明如何在為 Amazon EBS 加密指定客戶受管金鑰時，設定啟動執行個體所需的金鑰政策。

> **Note**
>
> AWS PCS 不需要額外的授權，即可使用預設值 AWS 受管金鑰 來保護您帳戶中的加密磁碟區。

目錄

- 範例 2：允許跨帳戶存取客戶受管金鑰的金鑰政策區段
- 在 AWS KMS 主控台編輯金鑰政策

概要

當 AWS PCS 啟動執行個體時，您可以使用下列項目 AWS KMS keys 進行 Amazon EBS 加密：

- AWS 受管金鑰 – Amazon EBS 在您的帳戶中建立、擁有和管理的加密金鑰。這是新帳戶的預設加密金鑰。除非您指定客戶受管金鑰，否則 Amazon EBS 會使用 AWS 受管金鑰 進行加密。
- 客戶受管金鑰 – 您建立、擁有和管理的自訂加密金鑰。如需詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的建立 KMS 金鑰。

> **Note**
>
> 金鑰必須是對稱的。Amazon EBS 不支援非對稱客戶受管金鑰。

當您建立加密快照或指定加密磁碟區的啟動範本，或選擇預設啟用加密時，您可以設定客戶受管金鑰。

設定金鑰政策

您的 KMS 金鑰必須具有金鑰政策，允許 AWS PCS 使用客戶受管金鑰加密的 Amazon EBS 磁碟區啟動執行個體。

使用此頁面的範例來設定金鑰政策，讓 AWS PCS 存取您的客戶受管金鑰。您可以在建立金鑰時或稍後修改客戶受管金鑰的金鑰政策。

金鑰政策必須具有下列陳述式：

- 允許 `Principal`元素中指定的 IAM 身分直接使用客戶受管金鑰的陳述式。它包含對金鑰執行 AWS KMS `Encrypt`、`Decrypt`、`GenerateDataKey*`、 `ReEncrypt*`和 `DescribeKey`操作的許可。
- 陳述式，允許 `Principal`元素中指定的 IAM 身分使用 `CreateGrant`操作來產生授予，將其自身許可的子集委派給與 AWS KMS 或其他主體整合 AWS 服務 的 。這允許其使用金鑰來代表您建立加密資源。

當您將新的政策陳述式新增至金鑰政策時，請勿變更政策中的任何現有陳述式。

如需詳細資訊，請參閱：

- AWS CLI 命令參考中的 create-key
- 《 AWS CLI 命令參考》中的 put-key-policy
- 《 AWS Key Management Service 開發人員指南》中的尋找金鑰 ID 和金鑰 ARN
- AWS PCS 的服務連結角色
- 《Amazon EBS 使用者指南》中的 Amazon EBS 加密
- AWS Key Management Service 《 AWS Key Management Service 開發人員指南》中的

範例 1：允許存取客戶受管金鑰的金鑰政策區段

將下列政策陳述式新增至客戶受管金鑰的金鑰政策。將範例 ARN 取代為
您AWSServiceRoleForPCS服務連結角色的 ARN。此範例政策提供 AWS PCS 服務連結角色
(AWSServiceRoleForPCS) 使用客戶受管金鑰的許可。

```
{
   "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
        }
     }
}
```

範例 2：允許跨帳戶存取客戶受管金鑰的金鑰政策區段

如果您在與 AWS PCS 叢集不同的帳戶中建立客戶受管金鑰，您必須使用授權搭配金鑰政策，以允許跨帳戶存取金鑰。

授予 金鑰的存取權

1. 將下列政策陳述式新增至客戶受管金鑰的金鑰政策。將範例 ARN 取代為另一個帳戶的 ARN。將 *111122223333* 取代 AWS 帳戶 為您要在其中建立 AWS PCS 叢集的 的實際帳戶 ID。這允許您為特定帳戶中的 IAM 使用者或角色授予許可，以使用下文的 CLI 命令為金鑰建立授權。根據預設，使用者無法存取 金鑰。

   ```
   {.
       "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. 從您要在其中建立 AWS PCS 叢集的帳戶，建立授予，將相關許可委派給 AWS PCS 服務連結角色。的值grantee-principal是服務連結角色的 ARN。的值key-id是金鑰的 ARN。

   下列 create-grant CLI 命令範例提供帳戶 *111122223333* AWSServiceRoleForPCS中名為 的服務連結角色許可，以使用帳戶 *444455556666* 中的客戶受管金鑰。

```
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**
>
> 提出請求的使用者必須具有使用 kms:CreateGrant動作的許可。

   下列 IAM 政策範例允許帳戶 *111122223333* 中的 IAM 身分 (使用者或角色)，為帳戶 *444455556666* 中的客戶受管金鑰建立許可。

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

如需在其他 AWS 帳戶中為 KMS 金鑰建立授權的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的「AWS KMS中的授權」。

> **Important**
>
> 指定為承授者主體的服務連結角色名稱必須是現有角色的名稱。建立授予之後，為了確保授予允許 AWS PCS 使用指定的 KMS 金鑰，請勿刪除並重新建立服務連結角色。

在 AWS KMS 主控台編輯金鑰政策

前幾節的範例僅顯示如何對金鑰政策新增陳述式，而這只是其中一個用來變更金鑰政策的方法。變更金鑰政策的最簡單方法是針對金鑰政策使用 AWS KMS 主控台的預設檢視，並將 IAM 身分（使用者或角色）設為適當金鑰政策的其中一個金鑰使用者。如需詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的使用 AWS Management Console 預設檢視。

> **Warning**
>
> 主控台的預設檢視政策陳述式包含對客戶受管金鑰執行 AWS KMS Revoke操作的許可。如果您撤銷授予您帳戶中客戶受管金鑰 AWS 帳戶 存取權的授予，則 中的使用者會 AWS 帳戶 失去加密資料和金鑰的存取權。

AWS Parallel Computing 服務 使用界面端點存取 (AWS PrivateLink)

您可以使用 在 VPC 和 AWS Parallel Computing 服務 () 之間 AWS PrivateLink 建立私有連線AWS PCS。您可以 AWS PCS 像在 VPC 中一樣存取 ，無需使用網際網路閘道、NAT 裝置、VPN 連接或 AWS Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址即可存取 AWS PCS。

您可以建立由 AWS PrivateLink提供支援的介面端點來建立此私有連線。我們會在您為介面端點啟用的每個子網中建立端點網路介面。這些是請求者管理的網路介面，可作為目的地為 AWS PCS之流量的進入點。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的AWS 服務 透過 存取 AWS PrivateLink 。

的考量事項 AWS PCS

在您設定 的介面端點之前 AWS PCS，請檢閱《 AWS PrivateLink 指南》中的使用介面 VPC 端點存取 AWS 服務。

AWS PCS 支援透過介面端點呼叫其所有 API 動作。

如果您的 VPC 沒有直接網際網路存取，您必須設定 VPC 端點，讓運算節點群組執行個體能夠呼叫 `RegisterComputeNodeGroupInstance` API AWS PCS 動作。

建立 的介面端點 AWS PCS

您可以使用 Amazon VPC AWS PCS 主控台或 AWS Command Line Interface () 建立 的介面端點AWS CLI。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的建立介面端點。

AWS PCS 使用下列服務名稱建立 的介面端點：

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

將##取代為 的 ID， AWS 區域 以在 中建立端點，例如 `us-east-1`。

如果您為介面端點啟用私有 DNS，您可以使用 AWS PCS 其預設的區域 DNS 名稱向 提出 API 請求。例如 `pcs.us-east-1.amazonaws.com`。

為您的介面端點建立端點政策

端點政策為 IAM 資源，您可將其連接至介面端點。預設端點政策允許 AWS PCS 透過介面端點完整存取 。若要控制允許 AWS PCS 從您的 VPC 存取 ，請將自訂端點政策連接至介面端點。

端點政策會指定以下資訊：

- 可執行動作 (AWS 帳戶、IAM 使用者和 IAM 角色) 的主體。
- 可執行的動作。

- 可供執行動作的資源。

如需詳細資訊，請參閱「AWS PrivateLink 指南」中的使用端點政策控制對服務的存取。

範例： AWS PCS 動作的 VPC 端點政策

以下是自訂端點政策的範例。當您將此政策連接至介面端點時，它會授予所有主體對具有指定 *cluster-id* 之叢集的所列 AWS PCS 動作的存取權。以叢集的 AWS 區域 ID 取代*##*，例如 us-east-1。將 *account-id* 取代為叢集的 AWS 帳戶 號碼。

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

AWS 平行運算服務的 Identity and Access Management

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （ 登入 ） 和授權 （ 具有許可 ） 來使用 AWS PCS 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- AWS Parallel Computing Service 如何與 IAM 搭配使用

- AWS Parallel Computing Service 的身分型政策範例
- AWSAWS Parallel Computing Service 的 受管政策
- AWS PCS 的服務連結角色
- AWS PCS 的 Amazon EC2 Spot 角色
- AWS PCS 的最低許可
- 平行運算服務的 IAM AWS 執行個體描述檔
- 平行 AWS 運算服務身分和存取的故障診斷

目標對象

使用方式 AWS Identity and Access Management (IAM) 會有所不同，取決於您在 AWS PCS 中執行的工作。

服務使用者 – 如果您使用 AWS PCS 服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS PCS 功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 AWS PCS 中的功能，請參閱 平行 AWS 運算服務身分和存取的故障診斷。

服務管理員 – 如果您在公司負責 AWS PCS 資源，您可能擁有 AWS PCS 的完整存取權。您的任務是判斷服務使用者應存取哪些 AWS PCS 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何搭配 AWS PCS 使用 IAM，請參閱 AWS Parallel Computing Service 如何與 IAM 搭配使用。

IAM 管理員 – 如果您是 IAM 管理員，建議您了解如何撰寫政策以管理 AWS PCS 存取的詳細資訊。若要檢視您可以在 IAM 中使用的 AWS PCS 身分型政策範例，請參閱 AWS Parallel Computing Service 的身分型政策範例。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的登入資料，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的登入資料以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的 IAM 中的AWS 多重要素驗證。

AWS 帳戶 根使用者

建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

根據最佳實務， 要求人類使用者，包括需要管理員存取權的使用者，使用 聯合身分提供者 AWS 服務來使用臨時登入資料來存取 。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、 AWS Directory Service、 Identity Center 目錄或任何使用透過身分來源提供的登入資料 AWS 服務 存取的使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色（主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以直接將政策連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。

- 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。
- 服務連結角色 – 服務連結角色是一種連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件，當與身分或資源相關聯時， AWS 會定義其許可。當委託人 （使用者、根使用者或角色工作階段） 發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 API AWS 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種服務，用於分組和集中管理您企業擁有 AWS 帳戶 的多個 。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的

許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 服務 的 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 (RCPs)。

- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS Parallel Computing Service 如何與 IAM 搭配使用

在您使用 IAM 管理 AWS PCS 的存取權之前，請先了解哪些 IAM 功能可與 AWS PCS 搭配使用。

您可以搭配 AWS 平行運算服務使用的 IAM 功能

IAM 功能	AWS PCS 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
主體許可	是
服務角色	否

IAM 功能	AWS PCS 支援
服務連結角色	是

若要全面了解 AWS PCS 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《AWS IAM 使用者指南》中的與 IAM 搭配使用的 服務。

AWS PCS 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

AWS PCS 的身分型政策範例

若要檢視 AWS PCS 身分型政策的範例，請參閱 AWS Parallel Computing Service 的身分型政策範例。

AWS PCS 內的資源型政策

支援資源型政策：否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當委託人和資源位於不同位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予委託人實體 （使用者或角色） 存取資源的許可。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

AWS PCS 的政策動作

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS PCS 動作清單，請參閱《服務授權參考》中的AWS 平行運算服務定義的動作。

AWS PCS 中的政策動作在動作之前使用以下字首：

```
pcs
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

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

AWS PCS 的政策資源

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

若要查看 AWS PCS 資源類型及其 ARNs，請參閱《服務授權參考》中的AWS 平行運算服務定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱AWS 平行運算服務定義的動作。

若要檢視 AWS PCS 身分型政策的範例，請參閱 AWS Parallel Computing Service 的身分型政策範例。

AWS PCS 的政策條件索引鍵

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值， 會使用邏輯OR操作 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定的條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AWS PCS 條件金鑰清單，請參閱《服務授權參考》中的AWS 平行運算服務的條件金鑰。 若要了解您可以使用條件金鑰的動作和資源，請參閱AWS 平行運算服務定義的動作。

若要檢視 AWS PCS 身分型政策的範例，請參閱 AWS Parallel Computing Service 的身分型政策範例。

AWS PCS ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 搭配 AWS PCS

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在 中 AWS，這些屬性稱為標籤。您可以將標籤連接至 IAM 實體 （使用者或角色） 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的使用屬性型存取控制 (ABAC)。

搭配 AWS PCS 使用臨時登入資料

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法運作。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的 使用 IAM。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的從使用者切換至 IAM 角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

AWS PCS 的跨服務主體許可

支援轉寄存取工作階段 (FAS)：是

當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合

AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱轉發存取工作階段。

AWS PCS 的服務角色

支援服務角色：否

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。

> **Warning**
>
> 變更服務角色的許可可能會中斷 AWS PCS 功能。只有在 AWS PCS 提供指引時，才能編輯服務角色。

AWS PCS 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理 AWS PCS 服務連結角色的詳細資訊，請參閱 AWS PCS 的服務連結角色。

AWS Parallel Computing Service 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 AWS PCS 資源的許可。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 AWS PCS 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的AWS 平行運算服務的動作、資源和條件金鑰。

主題

- 政策最佳實務
- 使用 AWS PCS 主控台
- 允許使用者檢視他們自己的許可

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AWS PCS 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 等使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用 AWS PCS 主控台

若要存取 AWS 平行運算服務主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AWS PCS 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

如需使用 AWS PCS 主控台所需最低許可的詳細資訊，請參閱 AWS PCS 的最低許可。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
    "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 Parallel Computing Service 的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受 AWS 管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。當新的 AWS 服務 啟動或新的 API 操作可供現有服務使用時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 受管政策：AWSPCSComputeNodePolicy

您可以將 AWSPCSComputeNodePolicy 連接至 IAM 實體。您可以將此政策連接至您指定的 AWS PCS 運算節點 IAM 角色，以允許使用該角色連線至 AWS PCS 叢集的節點。

AWS 當您使用主控台建立運算節點群組時，PCS 會將此政策連接至運算節點群組角色。

許可詳細資訊

此政策包含以下許可。

- `pcs:RegisterComputeNodeGroupInstance` – 允許 AWS PCS 運算節點 (EC2 執行個體） 向 AWS PCS 叢集註冊。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSPCSComputeNodePolicy。

AWS 受管政策：AWSPCSServiceRolePolicy

您無法將 AWSPCSServiceRolePolicy 連接至 IAM 實體。此政策會連接到服務連結角色，允許 AWS PCS 代表您執行動作。如需詳細資訊，請參閱AWS PCS 的服務連結角色。

許可詳細資訊

此政策包含以下許可。

- `ec2` – 允許 AWS PCS 建立和管理 Amazon EC2 資源。
- `iam` – 允許 AWS PCS 為 Amazon EC2 機群建立服務連結角色，並將角色傳遞給 Amazon EC2。

- `cloudwatch` – 允許 AWS PCS 將服務指標發佈至 Amazon CloudWatch。
- `secretsmanager` – 允許 AWS PCS 管理 AWS PCS 叢集資源的秘密。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSPCSServiceRolePolicy。

AWSAWS 受管政策的 PCS 更新

檢視自此服務開始追蹤這些變更以來 AWS ，PCS AWS 受管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 AWS PCS 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
AWSPCSComputeNodePolicy – 新政策	AWS PCS 新增了新的政策，以授予 AWS PCS 運算節點連線至 AWS PCS 叢集的許可。 AWS 當您在 PCS 主控台中建立運算節點群組時， AWS PCS 會將此政策連接至 IAM 角色。	2025 年 6 月 23 日
更新本文件中的 JSON	更正本文件中的 JSON 以包含 `"arn:aws:ec2:*:*:spot-instances-request/*"` 。	2024 年 9 月 5 日
AWS PCS 已開始追蹤變更	AWS PCS 開始追蹤其 AWS 受管政策的變更。	2024 年 8 月 28 日

AWS PCS 的服務連結角色

AWS 平行運算服務使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 AWS PCS 的唯一 IAM 角色類型。服務連結角色由 AWS PCS 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定 AWS PCS，因為您不必手動新增必要的許可。 AWS PCS 會定義其服務連結角色的許可，除非另有定義，否則只有 AWS PCS 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除角色的相關資源，才能刪除服務連結角色。這可保護您的 AWS PCS 資源，因為您不會意外移除存取資源的許可。

如需有關支援服務連結角色的其他 服務的資訊，請參閱AWS 使用 IAM 的服務，並在服務連結角色欄中尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

AWS PCS 的服務連結角色許可

AWS PCS 使用名為 AWSServiceRoleForPCS 的服務連結角色 – 授予 AWS PCS 管理 Amazon EC2 資源的許可。

AWSServiceRoleForPCS 服務連結角色信任下列服務擔任該角色：

- `pcs.amazonaws.com`

名為 AWSPCSServiceRolePolicy 的角色許可政策允許 AWS PCS 對特定資源完成動作。

您必須設定許可，以允許您的使用者、群組或角色建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

為 AWS PCS 建立服務連結角色

您不需要手動建立服務連結角色。 AWS PCS 會在您建立叢集時為您建立服務連結角色。

編輯 AWS PCS 的服務連結角色

AWS PCS 不允許您編輯 AWSServiceRoleForPCS 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱「IAM 使用者指南」的編輯服務連結角色。

刪除 AWS PCS 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

> **Note**
>
> 如果 AWS PCS 服務在您嘗試刪除資源時使用角色，則刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

移除 AWSServiceRoleForPCS 所使用的 AWS PCS 資源 AWSServiceRoleForPCS

您必須刪除所有叢集，才能刪除 AWSServiceRoleForPCS 服務連結角色。如需詳細資訊，請參閱刪除叢集。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForPCS 服務連結角色。如需詳細資訊，請參閱「IAM 使用者指南」中的刪除服務連結角色。

AWS PCS 服務連結角色支援的區域

AWS PCS 支援在所有提供服務的區域中使用服務連結角色。如需詳細資訊，請參閱AWS 區域與端點。

AWS PCS 的 Amazon EC2 Spot 角色

如果您想要建立使用 Spot 作為其購買選項的 AWS PCS 運算節點群組，您還必須在 中擁有 AWSServiceRoleForEC2Spot 服務連結角色 AWS 帳戶。您可以使用下列 AWS CLI 命令來建立角色。如需詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的建立服務連結角色和建立角色以委派許可給 AWS 服務。

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

> **Note**
>
> 如果您的 AWS 帳戶 已有 IAM `AWSServiceRoleForEC2Spot` 角色，則會收到下列錯誤。
>
> ```
> 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 的最低許可

本節說明 IAM 身分 （ 使用者、群組或角色 ） 使用服務所需的最低 IAM 許可。

內容

- 使用 API 動作的最小許可
- 使用標籤的最低許可

- 支援日誌的最低許可
- 服務管理員的最低許可

使用 API 動作的最小許可

API 動作	最低許可	主控台的其他許可
CreateCluster	`ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:GetSecurityGroupsForVpc, iam:CreateServiceLinkedRole, secretsmanager:CreateSecret, secretsmanager:TagResource, pcs:CreateCluster`	
ListClusters	`pcs:ListClusters`	
GetCluster	`pcs:GetCluster`	`ec2:DescribeSubnets`
DeleteCluster	`pcs:DeleteCluster`	
CreateComputeNodeGroup	`ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions,`	`iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster`

API 動作	最低許可	主控台的其他許可
	`ec2:DescribeInstanceTypes`, `ec2:DescribeInstanceTypeOfferings`, `ec2:RunInstances`, `ec2:CreateFleet`, `ec2:CreateTags`, `iam:PassRole`, `iam:GetInstanceProfile`, `pcs:CreateComputeNodeGroup`	
ListComputerNodeGroups	`pcs:ListComputeNodeGroups`	`pcs:GetCluster`
GetComputeNodeGroup	`pcs:GetComputeNodeGroup`	`ec2:DescribeSubnets`

API 動作	最低許可	主控台的其他許可
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 動作	最低許可	主控台的其他許可
DeleteQueue	`pcs:DeleteQueue`	

使用標籤的最低許可

在 AWS PCS 中將標籤與資源搭配使用需要下列許可。

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

支援日誌的最低許可

AWS PCS 會將日誌資料傳送至 Amazon CloudWatch Logs (CloudWatch Logs)。您必須確定您的身分具有使用 CloudWatch Logs 的最低許可。如需詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的管理 CloudWatch Logs 資源存取許可概觀。 Amazon CloudWatch

如需有關服務將日誌傳送至 CloudWatch Logs 所需許可的資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的啟用來自 AWS 服務的日誌記錄。

服務管理員的最低許可

下列 IAM 政策指定 IAM 身分 （使用者、群組或角色） 設定和管理 AWS PCS 服務所需的最低許可。

> **Note**
>
> 不設定和管理服務的使用者不需要這些許可。僅執行任務的使用者會使用安全 shell (SSH) 連線到叢集。 AWS Identity and Access Management (IAM) 不會處理 SSH 的身分驗證或授權。

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

平行運算服務的 IAM AWS 執行個體描述檔

在 EC2 執行個體上執行的應用程式必須在其提出的任何 AWS API 請求中包含 AWS 登入資料。我們建議您使用 IAM 角色來管理 EC2 執行個體上的臨時登入資料。您可以定義執行個體描述檔來執行此操作，並將其連接到您的執行個體。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Amazon EC2 的 IAM 角色。

> **Note**
>
> 當您使用 AWS Management Console 為 Amazon EC2 建立 IAM 角色時，主控台會自動建立執行個體描述檔，並為其提供與 IAM 角色相同的名稱。如果您使用 AWS CLI、 AWS API 動作或 AWS SDK 來建立 IAM 角色，您可以將執行個體描述檔建立為個別動作。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的執行個體描述檔。

建立運算節點群組時，您必須指定執行個體描述檔的 Amazon Resource Name (ARN)。您可以為部分或全部運算節點群組選擇不同的執行個體描述檔。

執行個體描述檔需求

執行個體描述檔 ARN

ARN 的 IAM 角色名稱部分必須以 開頭AWSPCS，或在路徑/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，請提供要包含在 ARN 路徑/aws-pcs/中的--path值iam create-instance-profile給 。例如：
>
> ```
> aws iam create-instance-profile --path /aws-pcs/ --instance-profile-name
>  example-role-2
> ```

許可

至少，AWS PCS 的執行個體描述檔必須包含下列政策。它可讓運算節點在 AWS PCS 服務運作時通知他們。

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

其他政策

您可以考慮將受管政策新增至執行個體描述檔。例如：

- AmazonS3ReadOnlyAccess 提供所有 S3 儲存貯體的唯讀存取權。
- AmazonSSMManagedInstanceCore 啟用 AWS Systems Manager 服務核心功能，例如直接從 Amazon 管理主控台進行遠端存取。
- CloudWatchAgentServerPolicy 包含在伺服器上使用 AmazonCloudWatchAgent 所需的許可。

您也可以包含自己的 IAM 政策，以支援您的特定使用案例。

建立執行個體描述檔

若要建立執行個體描述檔，您可以：

- 當您建立運算節點群組，讓 AWS PCS 使用最低必要政策為您建立基本設定檔時，請選取建立設定檔。
- 直接從 Amazon EC2 主控台建立執行個體描述檔。如需詳細資訊，請參閱AWS Identity and Access Management 《使用者指南》中的使用執行個體描述檔。

列出 AWS PCS 的執行個體描述檔

您可以使用下列 AWS CLI 命令，列出 AWS 區域 中符合 AWS PCS 名稱需求的執行個體描述檔。以適當的 取代 *us-east-1* AWS 區域。

```
aws iam list-instance-profiles --region us-east-1 --query "InstanceProfiles[?
starts_with(InstanceProfileName, 'AWSPCS') || contains(Path, '/aws-pcs/')].
[InstanceProfileName]" --output text
```

平行 AWS 運算服務身分和存取的故障診斷

使用以下資訊來協助您診斷和修正使用 AWS PCS 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 AWS PCS 中執行動作
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取我的 AWS PCS 資源

我無權在 AWS PCS 中執行動作

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 pcs:*GetWidget* 許可時發生。

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

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 pcs:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 iam:PassRole動作，您的政策必須更新，以允許您將角色傳遞至 AWS PCS。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為 的 IAM marymajor 使用者嘗試使用主控台在 AWS PCS 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 AWS PCS 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 AWS PCS 是否支援這些功能，請參閱 AWS Parallel Computing Service 如何與 IAM 搭配使用。
- 若要了解如何在您擁有 AWS 帳戶 的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中提供存取權給 IAM 使用者。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。

- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

AWS 平行運算服務的合規驗證

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- HIPAA 合格服務參考 – 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- AWS 合規資源 – 此工作手冊和指南的集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明跨多個架構 （包括國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 保護 AWS 服務和映射指南至安全控制的最佳實務。
- 《 AWS Config 開發人員指南》中的使用規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub – 這 AWS 服務 可讓您全面檢視其中的安全狀態 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱「Security Hub 控制參考」。
- Amazon GuardDuty – 這會監控您的環境是否有可疑和惡意活動，以 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險和符合法規和業界標準的方式。

AWS 平行運算服務的彈性

AWS 全球基礎設施是以 AWS 區域 和 可用區域為基礎建置。 AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高度備援聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

AWS 平行運算服務中的基礎設施安全性

作為受管服務， AWS Parallel Computing Service 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及 如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 AWS PCS。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

AWS PCS 建立叢集時，服務會在服務擁有的帳戶中啟動 Slurm 控制器，與您帳戶中的運算節點分開。為了橋接控制器與運算節點之間的通訊， AWS PCS 會在您的 VPC 中建立跨帳戶彈性網路界面 (ENI)。Slurm 控制器使用 ENI 來管理和與不同 的運算節點通訊 AWS 帳戶，維護資源的安全性和隔離，同時促進高效的 HPC 和 AI/ML 操作。

Parallel Computing Service AWS 中的漏洞分析和管理

組態和 IT 控制是 AWS 與您之間的共同責任。如需詳細資訊，請參閱 AWS 共同責任模型。 AWS 會處理服務帳戶中基礎基礎設施的基本安全任務，例如在控制器執行個體上修補作業系統、防火牆組態和 AWS 基礎設施災難復原。這些程序已由適當的第三方進行檢閱並認證。如需詳細資訊，請參閱安全性、身分和合規的最佳實務。

> **Note**
>
> 更新時無法使用 Slurm 控制器。執行中的任務不會受到影響。叢集的控制器無法使用時提交的任務會保留，直到控制器可用為止。

您有責任確保基礎基礎設施在 中的安全性 AWS 帳戶：

- 維護您的程式碼，包括更新和安全性修補程式。

- 修補和更新運算節點群組 Amazon Machine Image (AMI) 中的作業系統，並更新運算節點群組以使用更新的 AMI。
- 更新排程器，使其保持在支援的版本內。更新運算節點群組的 AMI，並更新運算節點群組以使用更新的 AMI。
- 驗證和加密使用者用戶端與其連線節點之間的通訊。

如需更新運算節點群組 AMI 的詳細資訊，請參閱 AWS PCS 的 Amazon Machine Image AMIs)。

預防跨服務混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多權限的實體執行該動作。在 中 AWS，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

我們建議在資源政策中使用 `aws:SourceArn`和 `aws:SourceAccount`全域條件內容索引鍵，以限制 AWS 平行運算服務 (AWS PCS) 為資源提供另一項服務的許可。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 `aws:SourceArn`。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

防範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容索引鍵，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 `aws:SourceArn` 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`。

如果 `aws:SourceArn` 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須使用這兩個全域條件內容索引鍵來限制許可。

的值`aws:SourceArn`必須是叢集 ARN。

下列範例示範如何在 AWS PCS 中使用 `aws:SourceArn`和 `aws:SourceAccount`全域條件內容金鑰，以防止混淆代理人問題。

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

作為運算節點群組一部分佈建之 Amazon EC2 執行個體的 IAM 角色

AWS PCS 會自動協調叢集中每個已設定運算節點群組的 Amazon EC2 容量。建立運算節點群組時，使用者必須透過 `iamInstanceProfileArn` 欄位提供 IAM 執行個體描述檔。執行個體描述檔會指定與佈建的 EC2 執行個體相關聯的許可。 AWS PCS 接受任何具有 `AWSPCS` 做為角色名稱字首或 `/aws-pcs/` 做為角色路徑一部分的角色。建立或更新運算節點群組的 IAM 身分（使用者或角色）需要`iam:PassRole`許可。當使用者呼叫 `CreateComputeNodeGroup`或 `UpdateComputeNodeGroup` API 動作時，AWS PCS 會檢查使用者是否可以執行`iam:PassRole`動作。

下列政策範例會授與僅傳遞名稱開頭為 `AWSPCS` 之 IAM 角色的許可。

```
{
    "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 Parallel Computing Service 的安全最佳實務

本節說明 AWS 平行運算服務 (AWS PCS) 特有的安全最佳實務。若要進一步了解 中的安全最佳實務 AWS，請參閱安全性、身分和合規的最佳實務。

AMI 相關安全性

- 請勿將 AWS PCS 範例 AMIs 用於生產工作負載。範例 AMIs 不受支援，且僅用於測試。
- 定期更新運算節點群組 AMI 中的作業系統和軟體，以緩解漏洞。
- 僅使用從官方 AWS 來源下載的已驗證官方 AWS PCS 套件。
- 定期更新運算節點群組 AMI 中的 AWS PCS 套件，並更新運算節點以使用更新的 AMI。請考慮自動化此程序，以將漏洞降至最低。

如需詳細資訊，請參閱AWS PCS 的自訂 Amazon Machine Image AMIs)。

Slurm Workload Manager 安全性

- 實作存取控制和網路限制，以保護 Slurm 控制和運算節點。僅允許信任的使用者和系統提交任務和存取 Slurm 管理命令。
- 使用 Slurm 的內建安全功能，例如 Slurm 身分驗證，以確保任務提交和通訊已進行身分驗證。
- 更新 Slurm 版本以維持順暢的操作和叢集支援。

> **Important**
>
> 任何使用已達到支援生命週期 (EOSL) 結束之 Slurm 版本的叢集都會立即停止。使用使用者指南頁面頂端的連結來訂閱 AWS PCS 文件 RSS 摘要，以在 Slurm 版本接近 EOSL 時收到通知。

如需詳細資訊，請參閱AWS PCS 中的 Slurm 版本。

監控和記錄

- 使用 Amazon CloudWatch Logs 和 AWS CloudTrail 來監控和記錄叢集和 中的動作 AWS 帳戶。使用資料進行疑難排解和稽核。

網路安全

- 在不同的 VPC 中部署 AWS PCS 叢集，以隔離您的 HPC 環境與其他網路流量。
- 使用安全群組和網路存取控制清單 ACLs) 來控制 AWS PCS 執行個體和子網路的傳入和傳出流量。
- 使用 AWS PrivateLink 或 VPC 端點來保持叢集與網路 AWS 內其他服務之間的 AWS 網路流量。如需詳細資訊，請參閱AWS Parallel Computing 服務 使用界面端點存取 (AWS PrivateLink)。

AWS PCS 的記錄和監控

監控是維護 AWS PCS 和其他 AWS 資源可靠性、可用性和效能的重要部分。AWS 提供下列監控工具來監看 AWS PCS、在發生錯誤時回報，以及適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- Amazon CloudWatch Logs 可讓您監控、存放和存取來自 Amazon EC2 執行個體、CloudTrail 及其他來源的日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。
- AWS CloudTrail 會擷取由您的帳戶或代表 AWS 您的帳戶發出的 API 呼叫和相關事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱「AWS CloudTrail 使用者指南」。

AWS PCS 中的任務完成日誌

任務完成日誌提供您平行 AWS 運算服務 (AWS PCS) 任務完成時的金鑰詳細資訊，無需額外費用。您可以使用其他 AWS 服務來存取和處理您的日誌資料，例如 Amazon CloudWatch Logs、Amazon Simple Storage Service (Amazon S3) 和 Amazon Data Firehose； AWS PCS 會記錄任務的中繼資料，例如下列項目。

- 任務 ID 和名稱
- 使用者和群組資訊
- 任務狀態 （ 例如 `COMPLETED`、`FAILED`、`CANCELLED`)
- 使用的分割區
- 時間限制
- 開始、結束、提交和合格時間
- 節點清單和計數
- 處理器計數
- 工作目錄

- 資源用量 (CPU、記憶體)
- 結束代碼
- 節點詳細資訊 (名稱、執行個體 IDs、執行個體類型)

內容

先決條件

管理 AWS PCS 叢集的 IAM 主體必須允許 `pcs:AllowVendedLogDeliveryForResource`動作。

下列範例 IAM 政策會授予所需的許可。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PcsAllowVendedLogsDelivery",
         "Effect": "Allow",
         "Action": ["pcs:AllowVendedLogDeliveryForResource"],
         "Resource": [
            "arn:aws:pcs:::cluster/*"
         ]
      }
   ]
}
```

設定任務完成日誌

您可以使用 AWS Management Console 或 為您的 AWS PCS 叢集設定任務完成日誌 AWS CLI。

AWS Management Console

使用主控台設定任務完成日誌

1. 開啟 AWS PCS 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要新增任務完成日誌的叢集。
4. 在叢集詳細資訊頁面上，選擇日誌索引標籤。
5. 在任務完成日誌下，選擇新增以在 CloudWatch Logs、Amazon S3 和 Firehose 之間新增最多 3 個日誌交付目的地。
6. 選擇更新日誌交付。

AWS CLI

使用 設定任務完成日誌 AWS CLI

1. 建立日誌交付目的地：

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

取代：

- *region* — 您要建立目的地 AWS 區域 的 ，例如 us-east-1
- *pcs-logs-destination* — 目的地的名稱
- *resource-arn* — CloudWatch Logs 日誌群組、S3 儲存貯體或 Firehose 交付串流的 Amazon Resource Name (ARN)。

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 PutDeliveryDestination。

2. 將 PCS 叢集設定為日誌交付來源：

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_JOBCOMP_LOGS
```

取代：

- *region* — 叢集 AWS 區域 的 ，例如 us-east-1
- *cluster-logs-source-name* — 來源的名稱
- *cluster-arn* — AWS PCS 叢集的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 PutDeliverySource。

3. 將交付來源連接至交付目的地：

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

取代：

- *region* — AWS 區域，例如 us-east-1
- *cluster-logs-source* — 交付來源的名稱
- *destination-arn* — 交付目的地的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 CreateDelivery。

如何尋找任務完成日誌

您可以在 CloudWatch Logs 和 Amazon S3. AWS PCS 中設定日誌目的地，並使用下列結構化路徑名稱和檔案名稱。

CloudWatch Logs

AWS PCS 使用 CloudWatch Logs 串流的下列名稱格式：

```
AWSLogs/PCS/cluster-id/jobcomp.log
```

例如：AWSLogs/PCS/pcs_abc123de45/jobcomp.log

Amazon S3

AWS PCS 會針對 S3 路徑使用下列名稱格式：

```
AWSLogs/account-id/PCS/region/cluster-id/jobcomp/year/month/day/hour/
```

例如：AWSLogs/111122223333/PCS/us-east-1/pcs_abc123de45/jobcomp/2025/06/19/11/

AWS PCS 對日誌檔案使用以下名稱格式：

```
PCS_jobcomp_year-month-day-hour_cluster-id_random-id.log.gz
```

例如：PCS_jobcomp_2025-06-19-11_pcs_abc123de45_04be080b.log.gz

任務完成日誌欄位

AWS PCS 會將任務完成日誌資料寫入為 JSON 物件。JSON 容器jobcomp會保留任務詳細資訊。下表說明jobcomp容器內的欄位。某些欄位僅在特定情況下存在，例如陣列任務或異質任務。

任務完成日誌欄位

名稱	範例值	必要	備註
job_id	11	是	永遠具有 值
user	"root"	是	永遠具有 值
user_id	0	是	永遠具有 值
group	"root"	是	永遠具有 值
group_id	0	是	永遠具有 值
name	"wrap"	是	永遠具有 值
job_state	"COMPLETED"	是	永遠具有 值
partition	"Hydra-MpiQueue-abcdef01-7"	是	永遠具有 值
time_limit	"UNLIMITED"	是	永遠存在，但可能是 "UNLIMITED"

名稱	範例值	必要	備註
start_time	"2025-06-19T10:58:57"	是	永遠存在，但可能是 "Unknown"
end_time	"2025-06-19T10:58:57"	是	永遠存在，但可能是 "Unknown"
node_list	"Hydra-MpiNG-abcdef01-2345-1"	是	永遠具有 值
node_cnt	1	是	永遠具有 值
proc_cnt	1	是	永遠具有 值
work_dir	"/root"	是	永遠存在，但可能是 "Unknown"
reservation_name	"weekly_maintenance"	是	永遠存在，但可能是空字串 ""
tres.cpu	1	是	永遠具有 值
tres.mem.val	600	是	永遠具有 值
tres.mem.unit	"M"	是	可以是 "M"或 "bb"
tres.node	1	是	永遠具有 值
tres.billing	1	是	永遠具有 值
account	"finance"	是	永遠存在，但可能是空字串 ""

名稱	範例值	必要	備註
qos	"normal"	是	永遠存在，但可能是空字串 ""
wc_key	"project_1"	是	永遠存在，但可能是空字串 ""
cluster	"unknown"	是	永遠存在，但可能是 "unknown"
submit_time	"2025-06-19T10:55:46"	是	永遠存在，但可能是 "Unknown"
eligible_time	"2025-06-19T10:55:46"	是	永遠存在，但可能是 "Unknown"
array_job_id	12	編號	只有在任務是陣列任務時才會出現
array_task_id	1	編號	只有在任務是陣列任務時才會出現
het_job_id	10	編號	只有在任務是異質任務時才會出現
het_job_offset	0	編號	只有在任務是異質任務時才會出現
derived_exit_code_status	0	是	永遠具有 值
derived_exit_code_signal	0	是	永遠具有 值
exit_code_status	0	是	永遠具有 值
exit_code_signal	0	是	永遠具有 值

名稱	範例值	必要	備註
node_details[0].name	"Hydra-MpiNG-abcdef01-2345-1"	編號	永遠存在，但node_details 可能是 "[]"
node_details[0].instance_id	"i-0abcdef01234567a"	編號	永遠存在，但node_details 可能是 "[]"
node_details[0].instance_type	"t4g.micro"	編號	永遠存在，但node_details 可能是 "[]"

任務完成日誌範例

下列範例顯示各種任務類型和狀態的任務完成日誌：

```
{ "jobcomp": { "job_id": 1, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:32:57",
 "end_time": "2025-06-19T16:33:03", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name":
 "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
 "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
 "submit_time": "2025-06-19T16:29:40", "eligible_time": "2025-06-19T16:29:41",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name":
 "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type":
 "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 2, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:33:13",
 "end_time": "2025-06-19T16:33:14", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name":
 "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
 "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
```

```
 "submit_time": "2025-06-19T16:33:13", "eligible_time": "2025-06-19T16:33:13",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name":
 "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type":
 "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 3, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T22:58:57", "end_time":
 "2025-06-19T22:58:57", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T22:55:46",
 "eligible_time": "2025-06-19T22:55:46", "derived_exit_code_status": 0,
 "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal":
 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id":
 "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 4, "user": "root", "user_id": 0, "group": "root",
 "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-
MpiQueue-abcdef01-7", "time_limit": "525600", "start_time": "2025-06-19T23:04:27",
 "end_time": "2025-06-19T23:04:27", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/root", "reservation_name":
 "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
 "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
 "submit_time": "2025-06-19T23:01:38", "eligible_time": "2025-06-19T23:01:38",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" }, { "name":
 "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def345abc67890", "instance_type":
 "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 5, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "FAILED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:00", "end_time":
 "2025-06-19T23:09:00", "node_list": "(null)", "node_cnt": 0, "proc_cnt": 0,
 "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val":
 1, "unit": "G" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key":
 "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:00", "eligible_time":
 "2025-06-19T23:09:00", "derived_exit_code_status": 0, "derived_exit_code_signal": 0,
 "exit_code_status": 0, "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 6, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:36",
 "end_time": "2025-06-19T23:09:36", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
```

```
 { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
 "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:35",
 "eligible_time": "2025-06-19T23:09:36", "het_job_id": 6, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 7, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:10:03",
 "end_time": "2025-06-19T23:10:03", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
 { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
 "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:10:03",
 "eligible_time": "2025-06-19T23:10:03", "het_job_id": 7, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 8, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
 "2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
 "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 9, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
 "2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
 "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
 "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 10, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
 "2025-06-19T23:12:24", "node_list":"Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
```

```
 "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 11, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
 "2025-06-19T23:12:24", "node_list":"Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 600, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
 "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
 "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 13, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:57", "end_time":
 "2025-06-19T23:47:58", "node_list":"Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
 "eligible_time": "2025-06-19T23:43:56" , "array_job_id": 12, "array_task_id": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 12, "user": "root", "user_id": 0, "group": "root", "group_id":
 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:58", "end_time":
 "2025-06-19T23:47:58", "node_list":"Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
 1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
 "eligible_time": "2025-06-19T23:43:56" , "array_job_id": 12, "array_task_id": 2,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
```


AWS PCS 中的排程器日誌

您可以設定 AWS PCS 將詳細的記錄資料從叢集排程器傳送至 Amazon CloudWatch Logs、Amazon Simple Storage Service (Amazon S3) 和 Amazon Data Firehose。這可協助監控和故障診斷。

內容

先決條件

管理 AWS PCS 叢集的 IAM 主體必須允許 `pcs:AllowVendedLogDeliveryForResource`動作。

下列範例 IAM 政策會授予所需的許可。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PcsAllowVendedLogsDelivery",
         "Effect": "Allow",
         "Action": ["pcs:AllowVendedLogDeliveryForResource"],
         "Resource": [
            "arn:aws:pcs:::cluster/*"
         ]
      }
   ]
}
```

設定排程器日誌

您可以使用 AWS Management Console 或 為您的 AWS PCS 叢集設定排程器日誌 AWS CLI。

AWS Management Console

使用主控台設定排程器日誌

1. 開啟 AWS PCS 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要新增排程器日誌的叢集。
4. 在叢集詳細資訊頁面上，選擇日誌索引標籤。

5. 在排程器日誌下，選擇新增以在 CloudWatch Logs、Amazon S3 和 Firehose 之間新增最多 3 個日誌交付目的地。
6. 選擇更新日誌交付。

AWS CLI

使用 設定排程器日誌 AWS CLI

1. 建立日誌交付目的地：

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

取代：

- *region* — 您要建立目的地 AWS 區域 的 ，例如 us-east-1
- *pcs-logs-destination* — 目的地的名稱
- *resource-arn* — CloudWatch Logs 日誌群組、S3 儲存貯體或 Firehose 交付串流的 Amazon Resource Name (ARN)。

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 PutDeliveryDestination。

2. 將 PCS 叢集設定為日誌交付來源：

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_SCHEDULER_LOGS
```

取代：

- *region* — 叢集 AWS 區域 的 ，例如 us-east-1
- *cluster-logs-source-name* — 來源的名稱
- *cluster-arn* — AWS PCS 叢集的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 PutDeliverySource。

3. 將交付來源連接至交付目的地：

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

取代：

- *region* — AWS 區域，例如 us-east-1
- *cluster-logs-source* — 交付來源的名稱
- *destination-arn* — 交付目的地的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 CreateDelivery。

排程器日誌串流路徑和名稱

AWS PCS 排程器日誌的路徑和名稱取決於目的地類型。

- CloudWatch Logs
  - CloudWatch Logs 串流遵循此命名慣例。

```
AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
```

Example

```
AWSLogs/PCS/abcdef0123/slurmctld_24.05.log
```

- S3 bucket (S3 儲存貯體)
  - S3 儲存貯體輸出路徑遵循此命名慣例：

```
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 物件名稱遵循此慣例：

```
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
```

排程器日誌記錄範例

AWS PCS 排程器日誌是結構化的。除了 Slurm 控制器程序發出的日誌訊息之外，還包含叢集識別符、排程器類型、主要和修補程式版本等欄位。請見此處範例。

```
{
    "resource_id": "s3431v9rx2",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1721230979,
    "log_level": "info",
    "log_name": "slurmctld",
    "scheduler_type": "slurm",
    "scheduler_major_version": "24.11",
    "scheduler_patch_version": "5",
    "node_type": "controller_primary",
    "message": "[2024-07-17T15:42:58.614+00:00] Running as primary controller\n"
}
```

使用 Amazon CloudWatch 監控 AWS 平行運算服務

Amazon CloudWatch 會定期從叢集收集指標，藉此監控 AWS 您的平行運算服務 (AWS PCS) 叢集運作狀態和效能。這些指標會保留，可讓您存取歷史資料，並隨著時間深入了解叢集的效能。

CloudWatch 也可讓您監控 AWS PCS 啟動的 EC2 執行個體，以符合您的擴展需求。雖然您可以檢查執行中執行個體的日誌，但 CloudWatch 指標和日誌記錄資料通常會在執行個體終止後刪除。不過，您可以使用 EC2 啟動範本在執行個體上設定 CloudWatch 代理程式，即使在執行個體終止後仍保留指標和日誌，從而實現長期監控和分析。

探索本節中的主題，進一步了解如何使用 CloudWatch 監控 AWS PCS。

主題

使用 CloudWatch 監控 AWS PCS 指標

您可以使用 Amazon CloudWatch 監控 AWS PCS 叢集運作狀態，這會從您的叢集收集資料，並將其轉換為近乎即時的指標。這些統計資料會保留 15 個月，讓您可以存取歷史資訊，並更深入了解叢集的效能。叢集指標會以 1 分鐘的期間傳送至 CloudWatch。如需 CloudWatch 的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的什麼是 Amazon CloudWatch？。

AWS PCS 會將下列指標發佈至 CloudWatch 中的 AWS/PCS 命名空間。它們具有單一維度 `ClusterId`。

名稱	描述	個單位
ActualCapacity	IdleCapacity + UtilizedCapacity	計數
CapacityUtilization	UtilizedCapacity/ActualCapacity	計數
DesiredCapacity	ActualCapacity + PendingCapacity	計數
IdleCapacity	正在執行但未配置給任務的執行個體計數	計數
UtilizedCapacity	正在執行並配置給任務的執行個體計數	計數

使用 Amazon CloudWatch 監控 AWS PCS 執行個體

AWS PCS 會視需要啟動 Amazon EC2 執行個體，以符合 PCS 運算節點群組中定義的擴展需求。您可以在使用 Amazon CloudWatch 執行時監控這些執行個體。您可以透過登入並使用互動式命令列工具來檢查執行中執行個體的日誌。不過，根據預設，CloudWatch 指標資料只會在執行個體終止後保留一段有限期間，而且執行個體日誌通常會與傳回執行個體的 EBS 磁碟區一起刪除。若要保留 PCS 終止後

啟動之執行個體的指標或記錄資料，您可以使用 EC2 啟動範本在執行個體上設定 CloudWatch 代理程式。本主題提供監控執行中執行個體的概觀，並提供如何設定持久性執行個體指標和日誌的範例。

監控執行中的執行個體

尋找 AWS PCS 執行個體

若要監控 PCS 啟動的執行個體，請尋找與叢集或運算節點群組相關聯的執行中執行個體。然後，在指定執行個體的 EC2 主控台中，檢查狀態和警示以及監控區段。如果已針對這些執行個體設定登入存取，您可以連線到這些執行個體，並檢查執行個體上的各種日誌檔案。如需識別哪些執行個體由 PCS 管理的詳細資訊，請參閱 在 AWS PCS 中尋找運算節點群組執行個體。

啟用詳細指標

根據預設，執行個體指標會每隔 5 分鐘收集一次。若要每隔一分鐘收集指標，請在運算節點群組啟動範本中啟用詳細的 CloudWatch 監控。如需詳細資訊，請參閱開啟詳細的 CloudWatch 監控。

設定持久性執行個體指標和日誌

您可以在執行個體上安裝和設定 Amazon CloudWatch 代理程式，以保留其指標和日誌。這包含三個主要步驟：

1. 建立 CloudWatch 代理程式組態。
2. 將組態存放在 PCS 執行個體可以擷取的位置。
3. 撰寫 EC2 啟動範本，安裝 CloudWatch 代理程式軟體、擷取您的組態，並使用組態啟動 CloudWatch 代理程式。

如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 CloudWatch 代理程式收集指標、日誌和追蹤，以及 搭配 AWS PCS 使用 Amazon EC2 啟動範本。 Amazon CloudWatch

建立 CloudWatch Agent 組態

在執行個體上部署 CloudWatch 代理程式之前，您必須產生 JSON 組態檔案，指定要收集的指標、日誌和追蹤。您可以使用精靈或使用文字編輯器手動建立組態檔案。此示範會手動建立組態檔案。

在已安裝 AWS CLI 的電腦上，使用下列內容建立名為 config.json 的 CloudWatch 組態檔案。您也可以使用下列 URL 下載檔案的副本。

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

備註

- 範例檔案中的日誌路徑適用於 Amazon Linux 2。如果您的執行個體將使用不同的基礎作業系統，請視需要變更路徑。
- 若要擷取其他日誌，請在 下新增其他項目`collect_list`。
- 中的值`{brackets}`是範本變數。如需支援變數的完整清單，請參閱《Amazon CloudWatch 使用者指南》中的手動建立或編輯 CloudWatch 代理程式組態檔案。 Amazon CloudWatch
- 您可以選擇省略 `logs`或 ，`metrics`如果您不想收集這些資訊類型。

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

此檔案會指示 CloudWatch 代理程式監控數個檔案，這些檔案有助於診斷執行個體引導、身分驗證和登入，以及其他故障診斷網域中的錯誤。其中包含：

- `/var/log/cloud-init.log` – 執行個體組態初始階段的輸出

- /var/log/cloud-init-output.log – 從執行個體組態期間執行的命令輸出
- /var/log/amazon/pcs/bootstrap.log – 從執行個體組態期間執行的 PCS 特定操作輸出
- /var/log/slurmd.log – 從 Slurm 工作負載管理員的協助程式 slurmd 輸出
- /var/log/messages – 來自核心、系統服務和應用程式的系統訊息
- /var/log/secure – 與身分驗證嘗試相關的日誌，例如 SSH、sudo 和其他安全事件

日誌檔案會傳送至名為 的 CloudWatch 日誌群組/PCSLogs/instances。日誌串流是執行個體 ID 和日誌檔案基本名稱的組合。日誌群組的保留時間為 30 天。

此外， 檔案會指示 CloudWatch 代理程式收集數個常見的指標，並依執行個體 ID 彙總它們。

儲存組態

CloudWatch 代理程式組態檔案必須存放在 PCS 運算節點執行個體可存取的位置。有兩種常見的方法可以執行此操作。您可以將其上傳至運算節點群組執行個體可透過其執行個體描述檔存取的 Amazon S3 儲存貯體，或者，您可以將其儲存為 Amazon Systems Manager 參數存放區中的 SSM 參數。

上傳至 S3 儲存貯體

若要將檔案存放在 S3 中，請使用以下 AWS CLI 命令。在執行 命令之前，請進行下列取代：

- 使用您自己的 S3*3 ######## amzn-s3-demo-bucket*

首先，（如果您有現有的儲存貯體，這是選用的），請建立儲存貯體以保留組態檔案。

```
aws s3 mb s3://amzn-s3-demo-bucket
```

接著，將檔案上傳至儲存貯體。

```
aws s3 cp ./config.json s3://amzn-s3-demo-bucket/
```

儲存為 SSM 參數

若要將檔案儲存為 SSM 參數，請使用以下命令。在執行 命令之前，請進行下列取代：

- 將 *region-code* 取代為您使用 AWS PCS 的 AWS 區域。
- （選用）將 *AmazonCloudWatch-PCS* 取代為 參數的自有名稱。請注意，如果您從 變更名稱的字首AmazonCloudWatch-，您將需要在節點群組執行個體描述檔中特別新增對 SSM 參數的讀取存取權。

```
aws ssm put-parameter \
   --region region-code \
   --name "AmazonCloudWatch-PCS" \
   --type String \
   --value file://config.json
```

撰寫 EC2 啟動範本

啟動範本的特定詳細資訊取決於您的組態檔案是存放在 S3 還是 SSM 中。

使用存放在 S3 中的組態

此指令碼會安裝 CloudWatch 代理程式、從 S3 儲存貯體匯入組態檔案，以及啟動 CloudWatch 代理程式。使用您自己的詳細資訊取代此指令碼中的下列值：

- *amzn-s3-demo-bucket* – 您的帳戶可從中讀取的 S3 儲存貯體名稱
- */config.json* – 相對於儲存組態的 S3 儲存貯體根目錄的路徑

```
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==--
```

節點群組的 IAM 執行個體描述檔必須能夠存取 儲存貯體。以下是上述使用者資料指令碼中儲存貯體的範例 IAM 政策。

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

另請注意，執行個體必須允許 S3 和 CloudWatch 端點的傳出流量。這可以使用安全群組或 VPC 端點來完成，具體取決於您的叢集架構。

使用存放在 SSM 中的組態

此指令碼會安裝 CloudWatch 代理程式、從 SSM 參數匯入組態檔案，以及啟動 CloudWatch 代理程式。使用您自己的詳細資訊取代此指令碼中的下列值：

- （選用）將 *AmazonCloudWatch-PCS* 取代為 參數的自有名稱。

```
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==--
```

節點群組的 IAM 執行個體政策必須連接 CloudWatchAgentServerPolicy。

如果您的參數名稱開頭不是 AmazonCloudWatch- ，您將需要在節點群組執行個體描述檔中特別新增對 SSM 參數的讀取存取權。以下是說明 *DOC-EXAMPLE-PREFIX* 字首的範例 IAM 政策。

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "CustomCwSsmMParamReadOnly",
      "Effect" : "Allow",
      "Action" : [
        "ssm:GetParameter"
      ],
      "Resource" : "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

另請注意，執行個體必須允許 SSM 和 CloudWatch 端點的傳出流量。這可以使用安全群組或 VPC 端點來完成，具體取決於您的叢集架構。

使用 記錄 AWS 平行運算服務 API 呼叫 AWS CloudTrail

AWS PCS 已與 整合 AWS CloudTrail，此服務提供使用者、角色或 AWS PCS AWS 服務所採取動作的記錄。CloudTrail 會將 AWS PCS 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 AWS PCS 主控台的呼叫，以及對 AWS PCS API 操作的程式碼呼叫。如果您建立線索，您可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 AWS PCS 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。使用 CloudTrail 收集的資訊，您可以判斷對 AWS PCS 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

AWS CloudTrail 中的 PCS 資訊

建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當活動在 AWS PCS 中發生時，該活動會記錄於 CloudTrail 事件，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 中檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱「使用 CloudTrail 事件歷史記錄檢視事件」。

若要持續記錄 中的事件 AWS 帳戶，包括 AWS PCS 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。追蹤會記錄 AWS 分割區中所有 區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並對其採取行動。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

CloudTrail 會記錄所有 AWS PCS 動作，並記錄在AWS 平行運算服務 API 參考中。例如，對 `CreateComputeNodeGroup`、`UpdateQueue` 以及 `DeleteCluster` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 請求是使用根還是 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

從 AWS PCS 了解 CloudTrail 日誌檔案項目

權杖是一種組態，能讓事件以日誌檔案的形式交付至您指定的 S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

下列範例顯示`CreateQueue`動作的 CloudTrail 日誌項目。

```
{
    "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"
}
```

AWS PCS 的端點和服務配額

下列各節說明 AWS 平行運算服務 (AWS PCS) 的端點和服務配額。服務配額，先前稱為限制，是 的服務資源或操作數量上限 AWS 帳戶。

您的 AWS 帳戶 具有每個 AWS 服務的預設配額。除非另有說明，否則每個配額都是區域特定規定。您可以請求提高某些配額，而其他配額無法提高。

如需詳細資訊，請參閱 AWS 一般參考中的 AWS 服務配額。

內容

服務端點

區域名稱	區域	端點	通訊協定
美國東部 (俄亥俄)	us-east-2	pcs.us-east-2.amazonaws.com pcs-fips.us-east-2.amazonaws.com pcs-fips.us-east-2.api.aws pcs.us-east-2.api.aws	HTTPS
美國東部 (維吉尼亞北部)	us-east-1	pcs.us-east-1.amazonaws.com pcs-fips.us-east-1.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
		pcs-fips.us-east-1.api.aws pcs.us-east-1.api.aws	
美國西部 (奧勒岡)	us-west-2	pcs.us-west-2.amazonaws.com pcs-fips.us-west-2.amazonaws.com pcs-fips.us-west-2.api.aws pcs.us-west-2.api.aws	HTTPS
亞太區域 (新加坡)	ap-southeast-1	pcs.ap-southeast-1.amazonaws.com pcs.ap-southeast-1.api.aws	HTTPS
亞太區域 (雪梨)	ap-southeast-2	pcs.ap-southeast-2.amazonaws.com pcs.ap-southeast-2.api.aws	HTTPS
亞太區域 (東京)	ap-northeast-1	pcs.ap-northeast-1.amazonaws.com pcs.ap-northeast-1.api.aws	HTTPS
歐洲 (法蘭克福)	eu-central-1	pcs.eu-central-1.amazonaws.com pcs.eu-central-1.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
歐洲 (愛爾蘭)	eu-west-1	pcs.eu-west-1.amazonaws.com pcs.eu-west-1.api.aws	HTTPS
歐洲 (倫敦)	eu-west-2	pcs.eu-west-2.amazonaws.com pcs.eu-west-2.api.aws	HTTPS
Europe (Stockholm)	eu-north-1	pcs.eu-north-1.amazonaws.com pcs.eu-north-1.api.aws	HTTPS
AWS GovCloud （美國東部）	us-gov-east-1	pcs.us-gov-east-1.amazonaws.com pcs-fips.us-gov-east-1.amazonaws.com pcs-fips.us-gov-east-1.api.aws pcs.us-gov-east-1.api.aws	HTTPS

區域名稱	區域	端點	通訊協定
AWS GovCloud （美國西部）	us-gov-west-1	pcs.us-gov-west-1.amazonaws.com pcs-fips.us-gov-west-1.amazonaws.com pcs-fips.us-gov-west-1.api.aws pcs.us-gov-west-1.api.aws	HTTPS

Service Quotas

名稱	預設	可調整	Description
叢集	5	是	每個叢集的數量上限 AWS 區域。

> **Note**
>
> 預設值是 設定的初始配額 AWS。這些預設值與實際套用的配額值和可能的服務配額上限不同。如需詳細資訊，請參閱《Service Quotas 使用者指南》中的 Service Quotas 術語。

這些服務配額列於 中的AWS 平行運算服務 (PCS)AWS Management Console。若要針對顯示為可調整的值請求增加配額，請參閱 Service Quotas 使用者指南中的請求增加配額。 Service Quotas

> **Important**
>
> 請記得檢查 中的目前 AWS 區域 設定 AWS Management Console。

內部配額

下列配額為內部配額，不可調整。

名稱	預設	可調整	Description
並行叢集建立	1	否	狀態中的叢集數量上限`Creating` AWS 區域。
每個叢集的運算節點群組	10	否	每個叢集的運算節點群組數目上限。
每個叢集的佇列數	10	否	每個叢集的佇列數量上限。

其他服務的相關配額 AWS

AWS PCS 使用其他 AWS 服務。這些服務的服務配額會影響您使用 AWS PCS。

影響 AWS PCS 的 Amazon EC2 服務配額

- Spot 執行個體請求
- 執行隨需執行個體
- 啟動範本
- 啟動範本版本
- Amazon EC2 API 請求

如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Amazon EC2 服務配額。

平行 AWS 運算服務中的問題疑難排解

下列主題提供指引，以針對您在 AWS PCS 中可能遇到的一些問題進行疑難排解。

主題

AWS PCS 中的 EC2 執行個體會在重新啟動後終止和取代

問題概觀

重新啟動運算節點群組中的 EC2 執行個體後， AWS PCS 會自動終止並取代執行個體。

為什麼會發生這種情況

AWS PCS 不支援執行個體重新啟動。如果重新啟動 EC2 執行個體， AWS PCS 會將執行個體視為運作狀態不佳，並加以取代。如果 AWS PCS 持續終止並取代您的執行個體，這可能是因為執行個體啟動後，有些項目會重新啟動您的執行個體。一些範例包括透過 EC2 執行個體上的自動化重新開機（例如修補後的自動重新開機）、EC2 執行個體外部的自動化（例如網路管理應用程式）、其他服務 AWS（例如 AWS Systems Manager)，或人員手動重新開機。

處理方式

您可以檢查 `slurmctld`或 `slurmd` 日誌，查看您的執行個體是否已重新啟動。如需詳細資訊，請參閱 AWS PCS 中的排程器日誌 和 使用 Amazon CloudWatch 監控 AWS PCS 執行個體。下列範例`slurmctld`日誌項目表示執行個體已重新啟動：

Example

```
[2024-09-12T06:42:50.393+00:00] validate_node_specs: Node Login-1 unexpectedly rebooted
 boot_time=1726123354 last response=1726123285
```

由於修補而重新啟動

套用修補程式後，通常需要重新啟動。請勿將修補程式直接套用至屬於 AWS PCS 運算節點群組的 EC2 執行個體。如果您必須修補 EC2 執行個體，您應該將修補程式套用至更新的 Amazon Machine Image (AMI)，並更新運算節點群組以使用更新的 AMI。 AWS PCS 為這些運算節點群組啟動的新 EC2 執行個體將使用更新的（修補）AMI。如需詳細資訊，請參閱AWS PCS 的自訂 Amazon Machine Image AMIs)。

AWS PCS 使用者指南的文件歷史記錄

下表說明 AWS PCS 文件的重要變更。

日期	變更	說明文件更新	API 版本已更新
2025 年 7 月 3 日	AWS 歐洲（倫敦）發行的 PCS	AWS PCS 現已在歐洲（倫敦）(eu-west-2) 推出。 CloudFormation 範本可在歐洲（倫敦）開始使用 AWS 區域。如需詳細資訊，請參閱使用 AWS CloudFormation 建立範例 AWS PCS 叢集及AWS CloudFormation 建立範例 AWS PCS 叢集的 範本。	N/A
2025 年 7 月 1 日	更新主控台指示	您現在可以在主控台中建立叢集和運算節點群組時，讓 AWS PCS 為您建立基本執行個體描述檔和安全群組。如需詳細資訊，請參閱： • 在 AWS 平行運算服務中建立叢集 • 在 AWS PCS 中建立運算節點群組 • 平行運算服務的 IAM AWS 執行個體描述檔	N/A
2025 年 6 月 23 日	新的受管政策：AWSPCSComputeNodePolicy	新增了新的受管政策，授予 AWS PCS 運算節點連線至 AWS PCS 叢集的許可。如需詳細資訊，請參	N/A

日期	變更	說明文件更新	API 版本已更新
		閱AWS 受管政策：AWSPCSComputeNodePolicy。	
2025 年 6 月 19 日	新主題：任務完成日誌	使用任務完成日誌，在任務完成時記錄任務的詳細資訊，無需額外費用。如需詳細資訊，請參閱AWS PCS 中的任務完成日誌。	N/A

日期	變更	說明文件更新	API 版本已更新
2025 年 6 月 18 日	AWS 中的 PCS 版本 AWS GovCloud (US)	AWS PCS 現在可在 AWS GovCloud （美國東部） (us-gov-east-1) 和 AWS GovCloud （美國西部） (us-gov-west-1) 中使用。 CloudFormation 範本可在 中開始使用 AWS GovCloud (US) Regions。如需詳細資訊，請參閱使用 AWS CloudFormation 建立範例 AWS PCS 叢集及AWS CloudFormation 建立範例 AWS PCS 叢集的 範本。 如需 中 AWS PCS 服務端點的詳細資訊 AWS GovCloud (US) Regions，請參閱 AWS PCS 的端點和服務配額。 如需 中差異的詳細資訊 AWS GovCloud (US) Regions，請參閱AWS GovCloud (US) 《使用者指南》AWS 中的 PCS AWS GovCloud (US)。	N/A
2025 年 6 月 18 日	更新的 PCS 代理程式	更新 AWS PCS 代理程式 1.2.1-1 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式。 >>>>>> 公有	N/A

日期	變更	說明文件更新	API 版本已更新
2025 年 5 月 15 日	新功能：會計	Slurm 24.11 或更新版本現在支援 Slurm 會計。如需詳細資訊，請參閱AWS PCS 中的低語會計。	AWS 開發套件：2025-05-15
2025 年 5 月 15 日	針對 Slurm 24.11 更新	更新 Slurm 24.11.5 支援的使用者指南。如需詳細資訊，請參閱下列內容： • AWS PCS 中的 Slurm 版本 • 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 • AWS PCS 範例 AMIs版本備註	N/A
2025 年 5 月 5 日	已更新 Slurm 版本常見問答集	更新 Slurm 版本常見問答集 (FAQ)，有關接近或超過生命週期結束 (EOL) 的 Slurm 版本。如需詳細資訊，請參閱有關 AWS PCS 中 Slurm 版本的常見問題。	N/A
2025 年 4 月 17 日	新主題：如何取得運算節點群組詳細資訊	了解如何取得 AWS PCS 運算節點群組的詳細資訊，例如其 ID、ARN 和 AMI ID。如需詳細資訊，請參閱在 AWS PCS 中取得運算節點群組詳細資訊。	N/A

日期	變更	說明文件更新	API 版本已更新
2025 年 4 月 2 日	已更新 Slurm 安裝程式	更新 Slurm 安裝程式 24.05.7-1 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式。	N/A
2025 年 3 月 28 日	新增運算節點群組和佇列數量上限的配額	針對每個叢集的運算節點群組數目上限和每個叢集的佇列數目上限，新增了不可調整的內部配額。如需詳細資訊，請參閱內部配額。	N/A
2025 年 3 月 14 日	已變更 CloudFormation 範本中的屬性索引鍵	`Id` 現在`TemplateId` 適用於 CloudFormation 範本中的 `CustomLaunchTemplate` 屬性。如需詳細資訊，請參閱 AWS PCS CloudFormation 範本的一部分 中的資源。	N/A
2025 年 3 月 13 日	新增 AWS PCS 代理程式和 Slurm 的版本資訊	新增了新主題，說明每個版本的 AWS PCS 代理程式的變更。如需詳細資訊，請參閱AWS PCS 代理程式版本。 已將詳細資訊新增至 Slurm 版本主題，該主題說明對 Slurm 的 AWS PCS 支援的重要支援日期和詳細版本備註。如需詳細資訊，請參閱AWS PCS 中的 Slurm 版本。	N/A

日期	變更	說明文件更新	API 版本已更新
2025 年 3 月 7 日	更新的 PCS 代理程式	更新 AWS PCS 代理程式 1.2.0-1 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式。	N/A
2025 年 2 月 3 日	新增使用 AWS CloudFormation 搭配 AWS PCS 的主題	已將主題新增至 使用者指南，提供如何使用 AWS CloudFormation 搭配 AWS PCS 的範例。本主題提供使用範例 CloudFormation 範本建立範例 AWS PCS 叢集的程序，並簡短說明該範本的各節。如需詳細資訊，請參閱AWS CloudFormation 和 AWS PCS 入門。	N/A
2024 年 12 月 18 日	針對 Slurm 24.05 更新	已更新 Slurm 24.05 支援的使用者指南。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式及AWS PCS 範例 AMIs版本備註。	N/A
2024 年 12 月 18 日	更新 Slurm 23.11 範例 AMIs NVIDIA 版本	已更新 Slurm 23.11 範例 AMIs中的 NVIDIA 驅動程式和 CUDA 版本。如需詳細資訊，請參閱AWS PCS 範例 AMIs版本備註。	N/A

日期	變更	說明文件更新	API 版本已更新
2024 年 12 月 17 日	已更新 Slurm 安裝程式	已更新 Slurm 安裝程式 23.11.10-3 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式。	N/A
2024 年 12 月 13 日	更新的 PCS 代理程式	已更新 AWS PCS 代理程式 1.1.1-1 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式。	N/A
2024 年 12 月 6 日	更新的 PCS 代理程式和 Slurm 安裝程式	更新 AWS PCS 代理程式 1.1.0-1 和 Slurm 安裝程式 23.11.10-2 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式。	N/A
2024 年 12 月 6 日	新增有關作業系統支援的主題	如需詳細資訊，請參閱AWS PCS 中支援的作業系統。	N/A
2024 年 11 月 8 日	重組使用者指南	我們重組了使用者指南，將主題帶到頂層、將一些主題移至自己的頁面，並將類似的主題分組在一起。	N/A

日期	變更	說明文件更新	API 版本已更新
2024 年 11 月 7 日	已更新 AMI 主題	更新 Slurm 23.11.10 和 libjwt 17.0 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式及步驟 3 – 安裝 Slurm。 簡化並更正 AMIs的版本備註。如需詳細資訊，請參閱AWS PCS 範例 AMIs版本備註。	N/A
2024 年 11 月 7 日	新增了有關搭配 AWS PCS 使用加密 EBS 磁碟區的新主題	新增主題，說明 AWS PCS 中加密 EBS 磁碟區所需的 KMS 金鑰政策。如需詳細資訊，請參閱在 AWS PCS 中與加密 EBS 磁碟區搭配使用所需的 KMS 金鑰政策。	N/A
2024 年 10 月 18 日	AWS PCS 代理程式 1.0.1-1 已發行	更新 AMI 相關文件以參考 AWS PCS 代理程式 1.0.1-1 版。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式及步驟 2 – 安裝 AWS PCS 代理程式。	N/A
2024 年 10 月 10 日	新增故障診斷章節	新增故障診斷章節，其中包含重新啟動後自動取代之 EC2 執行個體的相關主題。如需詳細資訊，請參閱平行 AWS 運算服務中的問題疑難排解。	N/A

日期	變更	說明文件更新	API 版本已更新
2024 年 9 月 23 日	更新使用 API 動作和服務管理員的最低許可	`CreateComputeNodeGroup` 和 `UpdateComputeNodeGroup` API 動作現在需要 `ec2:DescribeInstanceTypeOfferings` 許可。如需詳細資訊，請參閱AWS PCS 的最低許可。	N/A
2024 年 9 月 5 日	更新服務管理員最低許可的範例 IAM 政策	如需詳細資訊，請參閱服務管理員的最低許可。	N/A
2024 年 9 月 5 日	在 受管政策頁面中，將缺少的許可新增至 JSON	這只是文件的更正。實際受管政策並未變更。如需詳細資訊，請參閱AWSAWS Parallel Computing Service 的 受管政策。	N/A
2024 年 8 月 28 日	已新增受管政策頁面	如需詳細資訊，請參閱AWSAWS Parallel Computing Service 的 受管政策。	N/A
2024 年 8 月 28 日	AWS PCS 版本	AWS PCS 使用者指南的初始版本。	AWS SDK：2024-08-28

AWS 詞彙表

如需最新的 AWS 術語，請參閱 AWS 詞彙表 參考中的AWS 詞彙表。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。