AWS ParallelCluster 使用者指南（v2）

AWS ParallelCluster

AWS ParallelCluster: AWS ParallelCluster 使用者指南（v2）

Table of Contents

AWS ParallelCluster 是什麼

AWS ParallelCluster是AWS支援的開放原始碼叢集管理工具，可協助您在AWS 雲端. 它會自動設置所需的計算資源，調度程序和共享文件系統。您可以使AWS ParallelCluster用AWS Batch和Slurm排程器。

您可以使用AWS ParallelCluster快速建置和部署概念驗證和生產 HPC 運算環境。您還可以在其上構建和部署高級工作流程AWS ParallelCluster，例如可自動執行整個 DNA 測序工作流程的基因體入口網站。

定價

使用命AWS ParallelCluster令列介面 (CLI) 或 API 時，您只需為建立或更新AWS ParallelCluster映像和叢集時建立的AWS資源付費。如需詳細資訊，請參閱 AWS 所使用的 服務 AWS ParallelCluster。

設定 AWS ParallelCluster

主題

安裝 AWS ParallelCluster

AWS ParallelCluster 以 Python 套件的形式散佈pip，並使用 Python 套件管理員進行安裝。如需有關安裝 Python 套件的詳細資訊，請參閱 Python 封裝使用者指南中的安裝套件。

安裝方式 AWS ParallelCluster:

- 使用虛擬環境 (建議)
- 使用 pip

您可以在上的發行版本頁面CLI上找到最新版本的版本號碼 GitHub。

在這個指南中，命令範例假設您已安裝 Python v3。pip 命令範例使用的是 pip3 版本。

安裝 AWS ParallelCluster 在虛擬環境中 (建議使用)

我們建議您安裝 AWS ParallelCluster 在虛擬環境中。如果您在嘗試安裝時遇到問題 AWS ParallelCluster 使用pip3，您可以安裝 AWS ParallelCluster 在虛擬環境中隔離工具及其依賴關係。或者，您可以使用與平常不同的 Python 版本。

安裝 AWS ParallelCluster 在使用 pip 的非虛擬環境中

下列項目的主要分配方式 AWS ParallelCluster 在 Linux 上，視窗和 macOS 是pip，這是 Python 的軟件包管理器。它提供一個簡單的方法來安裝、升級和移除 Python 套件及其依存項目。

Current AWS ParallelCluster 版本

AWS ParallelCluster 定期更新。若要判斷您是否擁有最新版本，請參閱（詳見）的「發行版本」頁面 GitHub。

如果您已經擁有pip和受支持的 Python 版本，則可以安裝 AWS ParallelCluster 通過使用下面的命令。如果您已安裝 Python 版本 3+，我們建議您使用 **pip3** 命令。

```
$ pip3 install "aws-parallelcluster<3.0" --upgrade --user
```

安裝後要採取的步驟

安裝之後 AWS ParallelCluster，您可能需要將可執行文件路徑添加到您的PATH變量中。有關平台的特別說明，請參閱以下主題：

- Linux – 添加 AWS ParallelCluster 可執行到您的命令行路徑
- macOS – 添加 AWS ParallelCluster 可執行到您的命令行路徑
- Windows – 添加 AWS ParallelCluster 可執行到您的命令行路徑

您可以驗證 AWS ParallelCluster 通過運行正確安裝`pcluster version`。

```
$ pcluster version
2.11.9
```

AWS ParallelCluster 定期更新。若要更新至最新版本 AWS ParallelCluster，再次執行安裝命令。有關最新版本的詳細信息 AWS ParallelCluster，請參閱 AWS ParallelCluster 版本說明。

```
$ pip3 install "aws-parallelcluster<3.0" --upgrade --user
```

解除安裝 AWS ParallelCluster，使用`pip uninstall`。

```
$ pip3 uninstall "aws-parallelcluster<3.0"
```

如果您沒有 Python 和 `pip`，請使用適用於您環境的程序。

每個環境的詳細說明

- 安裝 AWS ParallelCluster 在虛擬環境中 (建議使用)
- 安裝 AWS ParallelCluster 在 Linux 上

- 安裝 AWS ParallelCluster 在 macOS 上
- 安裝 AWS ParallelCluster 在視窗上

安裝 AWS ParallelCluster 在虛擬環境中 (建議使用)

我們建議您安裝 AWS ParallelCluster 在虛擬環境中，以避免與其他pip軟件包的要求版本衝突。

必要條件

- 確認已安裝 pip 和 Python。我們建議pip3和 Python 3 版本 3.8。如果您使用的是 Python 2，請使用 pip 來取代 pip3，以及使用 virtualenv 來取代 venv。

若要安裝 AWS ParallelCluster 在虛擬環境中

1. 如果未安裝 virtualenv，請使用 pip3 安裝 virtualenv。如果 python3 -m virtualenv help 顯示說明資訊，請前往步驟 2。

   Linux, macOS, or Unix

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

   執行 exit 以離開目前的終端機視窗，並且開啟一個新的終端機視窗來套用環境的變更。

   Windows

   ```
   C:\>pip3 install --user --upgrade virtualenv
   ```

   執行 exit 以離開目前的命令提示，並且開啟新的命令提示以套用環境的變更。
2. 建立虛擬環境並為其命名。

   Linux, macOS, or Unix

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

   或者，您可以使用該 -p 選項以指定特定版本的 Python。

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

Windows

```
C:\>virtualenv %USERPROFILE%\apc-ve
```

3. 啟用新的虛擬環境。

   Linux, macOS, or Unix

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

   Windows

   ```
   C:\>%USERPROFILE%\apc-ve\Scripts\activate
   ```

4. 安裝 AWS ParallelCluster 進入您的虛擬環境。

   Linux, macOS, or Unix

   ```
   (apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster<3.0"
   ```

   Windows

   ```
   (apc-ve) C:\>pip3 install --upgrade "aws-parallelcluster<3.0"
   ```

5. 請確認 AWS ParallelCluster 已正確安裝。

   Linux, macOS, or Unix

   ```
   $ pcluster version
   2.11.9
   ```

   Windows

   ```
   (apc-ve) C:\>pcluster version
   2.11.9
   ```

您可以使用 `deactivate` 命令來離開虛擬環境。每次啟動工作階段時，都必須重新啟動環境。

若要升級至最新版本 AWS ParallelCluster，再次執行安裝命令。

Linux, macOS, or Unix

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster<3.0"
```

Windows

```
(apc-ve) C:\>pip3 install --upgrade "aws-parallelcluster<3.0"
```

安裝 AWS ParallelCluster 在 Linux 上

您可以安裝 AWS ParallelCluster 及其對大多數 Linux 發行版的依賴關係pip，通過使用 Python 的軟件包管理器。首先，判斷是否已安裝 Python 和 pip：

1. 若要判斷 Linux 版本是否包含 Python 和 pip，請執行 pip --version。

   ```
   $ pip --version
   ```

   如果您已pip安裝，請繼續安裝 AWS ParallelCluster 與點子主題。否則，請繼續步驟 2。
2. 若要判斷是否已安裝 Python，請執行 python --version。

   ```
   $ python --version
   ```

   如果你已經安裝了 Python 3 版本 3.6+ 或 Python 2 2.7 版本，請繼續安裝 AWS ParallelCluster 與點子主題。否則，請安裝 Python，然後返回此程序來安裝 pip。
3. 使pip用 Python 封裝授權單位提供的指令碼進行安裝。
4. 使用 curl 命令下載安裝指令碼。

   ```
   $ curl -O https://bootstrap.pypa.io/get-pip.py
   ```
5. 執行指令碼和 Python 來下載並安裝最新版本的 pip 及其他必要的支援套件。

   ```
   $ python get-pip.py --user
   ```

   或

   ```
   $ python3 get-pip.py --user
   ```

當您加入 `--user` 參數，指令碼會將 pip 安裝到路徑 `~/.local/bin`。

6. 若要確認包含的資料夾pip是否屬於PATH變數的一部分，請執行下列動作：

   a. 在您的使用者資料夾中尋找 Shell 的描述檔指令碼。如果您不確定您擁有哪個 Shell，請執行 `basename $SHELL`。

   ```
   $ ls -a ~
   .  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
   ```

   - Bash – `.bash_profile`、`.profile` 或 `.bash_login`
   - Zsh – `.zshrc`
   - Tcsh – `.tcshrc`、`.cshrc` 或 `.login`

   b. 在設定檔尾端新增匯出命令，類似於以下範例。

   ```
   export PATH=~/.local/bin:$PATH
   ```

   匯出命令會在現有 PATH 變數的前面插入路徑 (即這個範例中的 `~/.local/bin`)。

   c. 若要讓這些變更生效，請將描述檔重新載入到目前的工作階段。

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

7. 確認已正確安裝 pip。

   ```
   $ pip3 --version
   pip 21.3.1 from ~/.local/lib/python3.6/site-packages (python 3.6)
   ```

章節

- 安裝 AWS ParallelCluster 與 pip
- 添加 AWS ParallelCluster 可執行到您的命令行路徑
- 在 Linux 上安裝 Python

安裝 AWS ParallelCluster 與 **pip**

用pip於安裝 AWS ParallelCluster.

```
$ python3 -m pip install "aws-parallelcluster<3.0" --upgrade --user
```

當您使用`--user`開關時，請進行`pip`安裝 AWS ParallelCluster 到`~/.local/bin`。

請確認 AWS ParallelCluster 正確安裝。

```
$ pcluster version
2.11.9
```

更新到最新版本的 & CLI，再次執行安裝命令。

```
$ python3 -m pip install "aws-parallelcluster<3.0" --upgrade --user
```

添加 AWS ParallelCluster 可執行到您的命令行路徑

在使用 `pip` 進行安裝之後，您可能需要將 `pcluster` 可執行檔新增到作業系統的 `PATH` 環境變數。

若要驗證`pip`安裝的資料夾 AWS ParallelCluster，執行下列命令。

```
$ which pcluster
/home/username/.local/bin/pcluster
```

如果您在安裝時省略了`--user`開關 AWS ParallelCluster，可執行文件可能位於 Python 安裝的`bin`文件夾中。如果您不知道 Python 的安裝位置，請執行此命令。

```
$ which python
/usr/local/bin/python
```

請注意，輸出可能是符號連結的路徑，而非實際可執行檔。若要查看符號連結指向何處，請執行 `ls -al`。

```
$ ls -al $(which python)
/usr/local/bin/python -> ~/.local/Python/3.6/bin/python3.6
```

如果這是您在安裝 AWS ParallelCluster的步驟 3 中新增至路徑的同一個資料夾，則您已完成安裝。否則，您必須再次執行步驟 3a — 3c，將此額外資料夾新增至路徑。

在 Linux 上安裝 Python

如果您的發行版本不是 Python，或者附帶了早期版本，請在安裝之前安裝 `pip` Python AWS ParallelCluster.

在 Linux 上安裝 Python 3

1. 檢查是否已安裝 Python。

   ```
   $ python3 --version
   ```

   或

   ```
   $ python --version
   ```

   > **Note**
   >
   > 如果您的 Linux 發行版本隨附 Python，您可能需要安裝 Python 開發人員套件。開發人員套件包含編譯擴充功能和安裝所需的標頭和程式庫 AWS ParallelCluster。 使用您的套件管理員來安裝開發人員套件。它通常命名為 `python-dev` 或 `python-devel`。

2. 如果未安裝 Python 2.7 或更高版本，請將 Python 與您的分發套件管理工具一起安裝。命令和套件名稱有所不同：

   - 在 Debian 的衍生產品，例如 Ubuntu，使用 `apt`。

     ```
     $ sudo apt-get install python3
     ```

   - 在 Red Hat 和衍生產品，請使用 `yum`。

     ```
     $ sudo yum install python3
     ```

   - 在SUSE和衍生產品上，使用`zypper`。

     ```
     $ sudo zypper install python3
     ```

3. 若要驗證 Python 是否正確安裝，請開啟命令提示或 Shell，並執行以下命令。

   ```
   $ python3 --version
   ```

```
Python 3.8.11
```

安裝 AWS ParallelCluster 在 macOS 上

章節

必要條件

- Python 3 版本 3.7 + 或 Python 2 版本 2.7

請檢查 Python 安裝。

```
$ python --version
```

如果您的電腦尚未安裝 Python，或是您想要安裝不同版本的 Python，請遵從安裝 AWS ParallelCluster 在 Linux 上中的程序操作。

安裝 AWS ParallelCluster 在 macOS 上使用點子

您也可以pip直接使用安裝 AWS ParallelCluster。 如果您沒有pip，請按照主要安裝主題中的說明進行操作。執行 pip3 --version 查看您的 macOS 版本是否已包含 Python 和 pip3。

```
$ pip3 --version
```

若要安裝 AWS ParallelCluster 在 macOS 上

1. 從下載頁面下載並安裝最新版本的 Python。
2. 下載並執行 Python Packaging Authority 提供的 pip3 安裝指令碼。

   ```
   $ curl -O https://bootstrap.pypa.io/get-pip.py
   $ python3 get-pip.py --user
   ```

3. 使用新安裝pip3的安裝 AWS ParallelCluster。 如果您使用 Python 版本 3+，我們建議您使用命pip3令。

```
$ python3 -m pip install "aws-parallelcluster<3.0" --upgrade --user
```

4. 請確認 AWS ParallelCluster 已正確安裝。

```
$ pcluster version
2.11.9
```

如果找不到此程式，請將它新增到命令列路徑。

更新到最新版本的 & CLI，再次執行安裝命令。

```
$ pip3 install "aws-parallelcluster<3.0" --upgrade --user
```

添加 AWS ParallelCluster 可執行到您的命令行路徑

在使用 pip 進行安裝後，您可能需要將 pcluster 程式新增到作業系統的 PATH 環境變數中。程式的位置取決於 Python 的安裝位置。

Example AWS ParallelCluster 安裝位置-macOS 用 Python 3.6 和 **pip** (使用者模式)

```
~/Library/Python/3.6/bin
```

請以您的 Python 版本替代上述範例中的版本。

如果您不知道 Python 的安裝位置，請執行 which python。

```
$ which python3
/usr/local/bin/python3
```

輸出可能是符號連結的路徑，而非實際程式的路徑。執行 ls -al 來查看其指向的路徑。

```
$ ls -al /usr/local/bin/python3
lrwxr-xr-x  1 username  admin  36 Mar 12 12:47 /usr/local/bin/python3 -> ../Cellar/
python/3.6.8/bin/python3
```

pip 將程式安裝到包含 Python 應用程式的相同資料夾中。將此資料夾新增至 PATH 變數。

若要修改您的**PATH**變數 (Linux、macOS 或 Unix)

1. 在您的使用者資料夾中尋找 Shell 的描述檔指令碼。如果您不確定您擁有哪個 Shell，請執行 `echo $SHELL`。

   ```
   $ ls -a ~
   .  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
   ```

   - Bash – `.bash_profile`、`.profile` 或 `.bash_login`
   - Zsh – `.zshrc`
   - TCSH — `.tcshrc`、或 `.cshrc` `.login`
2. 將匯出命令新增至您的描述檔指令碼。

   ```
   export PATH=~/.local/bin:$PATH
   ```

   這個命令將路徑 (此範例中的 `~/.local/bin`) 新增到目前的 `PATH` 變數。
3. 將設定檔載入到目前工作階段中。

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

安裝 AWS ParallelCluster 在視窗上

您可以安裝 AWS ParallelCluster 在視窗上使用`pip`，這是 Python 的軟件包管理器。如果您已經有 `pip`，請遵循主要 安裝主題 中的指示操作。

章節

- 安裝 AWS ParallelCluster 使用 Python 和pip視窗
- 添加 AWS ParallelCluster 可執行到您的命令行路徑

安裝 AWS ParallelCluster 使用 Python 和**pip**視窗

Python 軟體基金會為 Windows 提供包含 `pip` 的安裝程式。

安裝 Python 和 **`pip`** (Windows)

1. 從下載頁面下載 Python 視窗 x86-64 安裝程式。

2. 執行安裝程式。
3. 選擇將 Python 3 添加到PATH。
4. 選擇 Install Now (立即安裝)。

安裝程式會將 Python 安裝到您的使用者資料夾中，並將其程式資料夾新增到您的使用者路徑。

若要安裝 AWS ParallelCluster 與 **pip3** (視窗)

如果您使用 Python 版本 3+，我們建議您使用 pip3 命令。

1. 從開始功能表打開命令提示符。
2. 使用下列命令確認 Python 和 pip 都已安裝正確。

```
C:\>py --version
Python 3.8.11
C:\>pip3 --version
pip 21.3.1 from c:\python38\lib\site-packages\pip (python 3.8)
```

3. 安裝 AWS ParallelCluster 使用pip.

```
C:\>pip3 install "aws-parallelcluster<3.0"
```

4. 請確認 AWS ParallelCluster 已正確安裝。

```
C:\>pcluster version
2.11.9
```

更新到最新版本的 & CLI，再次執行安裝命令。

```
C:\>pip3 install --user --upgrade "aws-parallelcluster<3.0"
```

添加 AWS ParallelCluster 可執行到您的命令行路徑

安裝後 AWS ParallelCluster 使用pip，將pcluster程式新增至作業系統的PATH環境變數。

您可以執行下列命令，找到 pcluster 程式的安裝位置。

```
C:\>where pcluster
```

```
C:\Python38\Scripts\pcluster.exe
```

如果該命令未傳回任何結果，則必須手動新增路徑。使用命令列或 Windows 檔案總管來探索它在您電腦上的安裝位置。典型路徑包括：

- Python 3 和 **pip3** – `C:\Python38\Scripts\`
- Python 3 和 **pip3**-用戶選項- `%APPDATA%\Python\Python38\Scripts`

> **Note**
>
> 包括版本號碼的資料夾名稱可能有所不同。前面的例子顯示了蟒蛇 38。視需要更換為您所使用的版本編號。

若要修改PATH變數 (視窗)

1. 按下 Windows 鍵並輸入 **environment variables**。
2. 選擇 Edit environment variables for your account (編輯您帳戶的環境變數)。
3. 選擇 PATH，然後選擇 [編輯]。
4. 新增路徑至 Variable value (變數值) 欄位。例如：***C:\new\path***
5. 選擇 OK (確定) 兩次以套用新的設定。
6. 關閉任何正在執行的命令提示，並重新開啟命令提示字元視窗。

設定 AWS ParallelCluster

安裝之後 AWS ParallelCluster，完成下列設定步驟。

驗證您的 AWS 帳戶的角色包含執行 `pcluster`CLI. 如需詳細資訊，請參閱AWS ParallelCluster 範例執行個體和使用者政策。

設定您的 AWS 認證。如需詳細資訊，請參閱配置 AWS CLI 中的 AWS CLI 用戶指南。

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

所以此 AWS 區域 啟動叢集的位置必須至少有一個 Amazon EC2 key pair。如需詳細資訊，請參閱 Amazon EC2使用者指南中的 Amazon EC2 金鑰對。

```
$ pcluster configure
```

設定精靈會提示您輸入建立叢集所需的所有資訊。使用時序列的細節不同 AWS Batch 作為調度程序與使用相比 Slurm。 如需叢集配置的詳細資訊，請參閱組態。

> **Note**
>
> 從版本 2.11.5 開始， AWS ParallelCluster 不支持使用 SGE 或 Torque 排程器。您可以在 2.11.4 以下版本中繼續使用它們，但它們不符合 future 更新或疑難排解支援的資格 AWS 服務及 AWS Support 團隊。

Slurm

從有效列表 AWS 區域 識別碼，選擇 AWS 區域 您希望叢集執行的位置。

> **Note**
>
> 的列表 AWS 區域 顯示的是基於您帳戶的分區，並且僅包括 AWS 區域 已針對您的帳戶啟用。如需有關啟用的詳細資訊 AWS 區域 對於您的帳戶，請參閱管理 AWS 區域 中的 AWS 一般參考。 顯示的例子來自 AWS 全域磁碟分割。如果您的帳戶在 AWS GovCloud (US) 分割區，僅 AWS 區域 在該分區中列出（gov-us-east-1和gov-us-west-1）。同樣，如果您的帳戶位於 AWS 中國分區，只cn-northwest-1有cn-north-1和顯示。如需完整清單 AWS 區域 支持 AWS ParallelCluster，請參閱支援地區。

```
Allowed values for the AWS ## ID:
1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ca-central-1
9. eu-central-1
```

```
10. eu-north-1
11. eu-south-1
12. eu-west-1
13. eu-west-2
14. eu-west-3
15. me-south-1
16. sa-east-1
17. us-east-1
18. us-east-2
19. us-west-1
20. us-west-2
AWS ## ID [ap-northeast-1]:
```

選擇要與叢集搭配使用的排程器。

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

選擇作業系統。

```
Allowed values for Operating System:
1. alinux2
2. centos7
3. ubuntu1804
4. ubuntu2004
Operating System [alinux2]:
```

> **Note**
>
> 在中加入`alinux2`了 Support AWS ParallelCluster 版本為 2.6.0。

輸入運算節點叢集的最小和最大大小。這是以執行個體數量為單位測量而得。

```
Minimum cluster size (instances) [0]:
Maximum cluster size (instances) [10]:
```

即會輸入標頭和運算節點執行個體類型。對於執行個體類型，您的帳戶執行個體限制足以滿足您的需求。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的隨需執行個體限制。

```
Master instance type [t2.micro]:
Compute instance type [t2.micro]:
```

key pair 是從所選 Amazon 註冊的密鑰對EC2中選擇 AWS 區域.

```
Allowed values for EC2 Key Pair Name:
1. prod-uswest1-key
2. test-uswest1-key
EC2 Key Pair Name [prod-uswest1-key]:
```

完成上述步驟後，決定是使用現有的VPC還是 let AWS ParallelCluster 為您創VPC建一個。如果您沒有正確配置VPC， AWS ParallelCluster 可以創建一個新的。它可以在同一個公有子網路中同時使用頭節點和計算節點，或者只使用公有子網路中的頭節點，其中包含私有子網路中的所有節點。它有可能達到你的VPCs數量限制 AWS 區域。 每個預設限制VPCs為五 AWS 區域。 如需有關此限制以及如何請求提高的詳細資訊，請參閱 Amazon VPC 使用者指南中的VPC和子網路。

如果你讓 AWS ParallelCluster 建立時VPC，您必須決定是否所有節點都應位於公有子網路中。

> **Important**
>
> VPCs創建者 AWS ParallelCluster 依預設，不要啟用VPC流程記錄。VPC流程記錄可讓您擷取有關進出VPCs. 如需詳細資訊，請參閱 Amazon VPC 使用者指南中的VPC流程日誌。

> **Note**
>
> 如果你選擇 `1. Master in a public subnet and compute fleet in a private subnet` AWS ParallelCluster 即使您指定了免費方案資源，也會建立會產生額外費用的NAT閘道。

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Network Configuration:
1. Master in a public subnet and compute fleet in a private subnet
2. Master and compute fleet in the same public subnet
Network Configuration [Master in a public subnet and compute fleet in a private
 subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
 finalized
```

如果您不建立新的VPC，則必須選取現有的VPC。

如果您選擇擁有 AWS ParallelCluster 創建VPC，記下 VPC ID，以便您可以使用 AWS CLI 稍後刪除它。

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

選取之VPC後，您需要決定是使用現有子網路還是建立新的子網路。

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

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

AWS Batch

從有效列表 AWS 區域 識別碼，選擇 AWS 區域 您希望叢集執行的位置。

```
Allowed values for AWS ## ID:
1. ap-northeast-1
2. ap-northeast-2
3. ap-south-1
4. ap-southeast-1
5. ap-southeast-2
6. ca-central-1
7. eu-central-1
8. eu-north-1
9. eu-west-1
10. eu-west-2
11. eu-west-3
12. sa-east-1
13. us-east-1
14. us-east-2
15. us-west-1
16. us-west-2
```

```
AWS ## ID [ap-northeast-1]:
```

選擇要與叢集搭配使用的排程器。

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

當選取 awsbatch 做為排程器時，alinux2 會用作為作業系統。

輸入運算節點叢集的最小和最大大小。這是在測量vCPUs。

```
Minimum cluster size (vcpus) [0]:
Maximum cluster size (vcpus) [10]:
```

已輸入頭節點實例類型。使用 awsbatch 排程器時，運算節點會使用 optimal 的執行個體類型。

```
Master instance type [t2.micro]:
```

Amazon EC2 key pair 是從所選 Amazon 註冊的密鑰對EC2中選擇的 AWS 區域.

```
Allowed values for EC2 Key Pair Name:
1. prod-uswest1-key
2. test-uswest1-key
EC2 Key Pair Name [prod-uswest1-key]:
```

決定是否使用現有的VPCs還是讓 AWS ParallelCluster VPCs為您創造。如果您沒有正確配置 VPC， AWS ParallelCluster 可以創建一個新的。它可以在同一個公有子網路中同時使用頭節點和計算節點，或者只使用公有子網路中的頭節點，其中包含私有子網路中的所有節點。它有可能達到你的VPCs數量限制 AWS 區域。 預設數目VPCs為 5。如需有關此限制以及如何請求提高的詳細資訊，請參閱 Amazon VPC 使用者指南中的VPC和子網路。

> **Important**
>
> VPCs創建者 AWS ParallelCluster 依預設，不要啟用VPC流程記錄。VPC流程記錄可讓您擷取有關進出VPCs. 如需詳細資訊，請參閱 Amazon VPC 使用者指南中的VPC流程日誌。

如果你讓 AWS ParallelCluster 創建一個VPC，決定是否所有節點都應該在公共子網中。

> **Note**
>
> 如果你選擇 `1. Master in a public subnet and compute fleet in a private subnet` AWS ParallelCluster 即使您指定了免費方案資源，也會建立會產生額外費用的NAT閘道。

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Network Configuration:
1. Master in a public subnet and compute fleet in a private subnet
2. Master and compute fleet in the same public subnet
Network Configuration [Master in a public subnet and compute fleet in a private
 subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
 finalized
```

如果您不建立新的VPC，則必須選取現有的VPC。

如果您選擇擁有 AWS ParallelCluster 創建VPC，記下 VPC ID，以便您可以使用 AWS CLI 稍後刪除它。

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

選取之VPC後，請決定是使用現有子網路還是建立新的子網路。

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

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

當您完成上述步驟後，一個簡單的叢集會啟動到VPC. VPC使用支援公用 IP 位址的現有子網路。子網路的路由表是`0.0.0.0/0 => igw-xxxxxx`。請注意下列條件：

- 必VPC須具有DNS Resolution = yes和DNS Hostnames = yes。
- VPC應該還具有正確domain-name的DHCP選項 AWS 區域。 預設DHCP選項集已指定必要的 AmazonProvidedDNS。 如果指定多個網域名稱伺服器，請參閱 Amazon VPC 使用者指南中的DHCP選項集。使用私有子網路時，請使用NAT閘道或內部 Proxy 來啟用計算節點的 Web 存取。如需詳細資訊，請參閱網路組態。

當所有設定都包含有效值時，您可以執行建立命令來啟動叢集。

```
$ pcluster create mycluster
```

叢集達到 "CREATE_COMPLETE" 狀態後，您可以使用一般用SSH戶端設定來連線至叢集。如需連接至 Amazon EC2 執行個體的詳細資訊，請參閱 Amazon EC2使用者指南中的EC2使用者指南。

若要刪除叢集，請執行下列命令。

```
$ pcluster delete --region us-east-1 mycluster
```

若要刪除中的網路資源VPC，您可以刪除網 CloudFormation 路堆疊。堆疊名稱開頭為」parallelclusternetworking-「，並以" YYYYMMDDHHMMSS "格式包含建立時間。您可以使用列表堆棧命令列出堆棧。

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

堆棧可以使用刪除堆棧命令被刪除。

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

為您創pcluster configure建的VPC不是在 CloudFormation 網絡堆棧中創建的。您可以在主控台中VPC手動刪除，或使用 AWS CLI.

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

最佳實務

最佳做法：選取主要執行個體類型

雖然主節點不執行任何工作，但其功能及其大小對叢集的整體效能至關重要。

選擇要用於主節點的執行個體類型時，您要評估下列項目：

- 叢集大小：主節點會協調叢集的擴展邏輯，並負責將新節點附加到排程器。如果您需要擴展和縮小大量節點的集群，那麼您想要為主節點提供一些額外的計算容量。
- 共用檔案系統：使用共用檔案系統在計算節點和主節點之間共用成品時，會考慮到主節點是暴露NFS伺服器的節點。因此，您想要選擇具有足夠網路頻寬和足夠專用 Amazon EBS 頻寬的執行個體類型來處理工作流程。

最佳做法：網路效能

有三個提示涵蓋了改善網絡通信的整個範圍的可能性。

- 置放群組：叢集置放群組是單一可用區域內執行個體的邏輯群組。如需置放群組的詳細資訊，請參閱 Amazon EC2 使用者指南中的放置群組。您可以將叢集配置為搭配`placement_group = your-placement-group-name`或 let 使用您自己的置放群組 AWS ParallelCluster 使用`"compute"`策略建立放置群組`placement_group = DYNAMIC`。如需詳細資訊，請參閱`placement_group`多重佇列模式與`placement_group`單一佇列模式的相關資訊。
- 增強型聯網：請考慮選擇支援增強型聯網的執行個體類型。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Linux 上的增強型聯網。
- 彈性網狀架構介面卡：若要支援高層級的可擴充執行個體間通訊，請考慮為您的EFA網路選擇網路介面。客製化EFA的作業系統 (OS) 略過硬體可增強執行個體間的通訊，並具有依需求的彈性和彈性 AWS 雲端。若要設定單一 Slurm 要使用的叢集佇列EFA，設定`enable_efa = true`。如需有關EFA搭配使用的詳細資訊 AWS ParallelCluster」，請參閱Elastic Fabric Adapter和`enable_efa`。如需有關的詳細資訊EFA，請參閱 Amazon Linux 執行個體EC2使用者指南中的彈性網狀架構介面卡。
- 執行個體頻寬：頻寬會隨執行個體大小擴展，請考慮選擇更適合您需求的執行個體類型，請參閱 Amazon EC2 使用者指南中的 Amazon EBS 優化執行個體和 Amazon EBS 磁碟區類型。

最佳做法：預算提醒

若要管理 AWS ParallelCluster 資源成本，我們建議您使用 AWS Budgets 建立預算及定義預算臨界值警示的作業 AWS 的費用。如需詳細資訊，請參閱 AWS Budgets 使用者指南。您也可以使用 Amazon CloudWatch 建立帳單警示。如需詳細資訊，請參閱建立帳單警示以監控您的估計 AWS 收費。

最佳做法：將叢集移至新叢集 AWS ParallelCluster 次要或修補程式版本

目前每個 AWS ParallelCluster 次要版本與其一起是獨立的`pcluster`CLI。若要將叢集移至新的次要或修補程式版本，您必須使用新版本的重新建立叢集。CLI

若要最佳化將叢集移至新次要版本的程序，或基於其他原因儲存共用儲存體資料，建議您使用下列最佳作法。

- 將個人數據保存在外部卷中，例如 Amazon EFS 和 FSx Lustre。通過這樣做，您可以輕鬆地將數據從一個集群移動到另一個集群。
- 使用建立下列類型的共用儲存系統 AWS CLI 或 AWS Management Console:
  - `[ebs]` 區段
  - `[efs]` 區段
  - `[fsx]` 區段

  將它們作為現有檔案系統新增至新的叢集配置。如此一來，當您刪除叢集時，它們會被保留，並且可以連接到新叢集。無論共用儲存系統是連接或從叢集中分離，通常都會產生費用。

  建議您將 Amazon 或 Amazon 用FSx於 Lustre 檔案系統EFS，因為這些檔案系統可以同時連接到多個叢集，而且您可以在刪除舊叢集之前將它們連接到新叢集。有關更多信息，請參閱 Amazon EFS 用戶指南中的安裝 Amazon EFS 文件系統和在 Amazon FSx Lustre Lustre 用戶指南中的訪問 FSx Lustre 文件系統。
- 使用自訂啟動程序動作來自訂執行個體，而非自訂執行個體AMI。這樣可以最佳化建立程序，因為AMI不需要為每個新版本建立新的自訂。
- 推薦的順序。
  1. 更新叢集配置以使用現有的檔案系統定義。
  2. 驗證`pcluster`版本並在需要時進行更新。
  3. 建立並測試新叢集。
     - 確定您的資料可在新叢集中使用。
     - 確定您的應用程式可在新叢集中運作。

4. 如果您是新叢集已經過完整測試和操作，而且您確定您不會使用舊叢集，請將其刪除。

從移動 CfnCluster 到 AWS ParallelCluster

AWS ParallelCluster 是的增強版本 CfnCluster。

如果您目前使用 CfnCluster，我們建議您使用 AWS ParallelCluster 而是使用它創建新的集群。即使您可以繼續使用 CfnCluster，但它不再被開發，也不會添加新的特性或功能。

和之間的主要區 CfnCluster 別 AWS ParallelCluster 在以下各節中說明。

AWS ParallelCluster CLI管理一組不同的叢集

使用建立的叢集`cfncluster`CLI無法使用 `pcluster`CLI. 下列命令不適用於由以下人員建立的叢集 CfnCluster：

```
pcluster list
pcluster update cluster_name
pcluster start cluster_name
pcluster status cluster_name
```

若要管理使用建立的叢集 CfnCluster，您必須使用 `cfncluster`CLI.

如果您需要 CfnCluster 套件來管理舊叢集，我們建議您從 Python 虛擬環境安裝並使用它。

AWS ParallelCluster 並 CfnCluster 使用不同的IAM自定義策略

先前用於 CfnCluster 叢集建立的自訂IAM原則無法搭配使用 AWS ParallelCluster。 如果您需要自訂原則 AWS ParallelCluster，您必須建立新的。請參閱 AWS ParallelCluster 指南。

AWS ParallelCluster 並 CfnCluster 使用不同的配置文件

所以此 AWS ParallelCluster 組態檔位於`~/.parallelcluster`資料夾中。組 CfnCluster態檔案位於`~/.cfncluster`資料夾中。

如果您想要使用現有的 CfnCluster 組態檔案 AWS ParallelCluster，則您必須完成下列動作：

1. 將組態檔從 `~/.cfncluster/config` 移至 `~/.parallelcluster/config`。
2. 如果您使用 `extra_json` 組態參數，請將其變更為如下所示。

CfnCluster 設置：

```
extra_json = { "cfncluster" : { } }
```

AWS ParallelCluster 設置：

```
extra_json = { "cluster" : { } }
```

在 AWS ParallelCluster，神經節預設為停用

In (入) AWS ParallelCluster，神經節預設為停用。要啟用神經節，請完成以下步驟：

1. 設定 `extra_json` 參數，如下所示：

   ```
   extra_json = { "cluster" : { "ganglia_enabled" : "yes" } }
   ```

2. 變更磁頭安全性群組以允許連線至連接埠 80。

   必須透過新增安全群組規則來修改 `parallelcluster-<CLUSTER_NAME>-MasterSecurityGroup-<xxx>` 安全群組，以允許公有 IP 到連接埠 80 的傳入連線。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的將規則新增至安全群組。

支援地區

AWS ParallelCluster 版本 2.x 可在以下 AWS 區域:

區域名稱	區域
美國東部 (俄亥俄)	us-east-2
美國東部 (維吉尼亞北部)	us-east-1
美國西部 (加利佛尼亞北部)	us-west-1
美國西部 (奧勒岡)	us-west-2
非洲 (開普敦)	af-south-1
亞太區域 (香港)	ap-east-1
亞太區域 (孟買)	ap-south-1

區域名稱	區域
亞太區域 (首爾)	ap-northeast-2
亞太區域 (新加坡)	ap-southeast-1
亞太區域 (雪梨)	ap-southeast-2
亞太區域 (東京)	ap-northeast-1
加拿大 (中部)	ca-central-1
中國 (北京)	cn-north-1
中國 (寧夏)	cn-northwest-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 (米蘭)	eu-south-1
歐洲 (巴黎)	eu-west-3
歐洲 (斯德哥爾摩)	eu-north-1
中東 (巴林)	me-south-1
南美洲 (聖保羅)	sa-east-1
AWS GovCloud (美國東部)	us-gov-east-1
AWS GovCloud (美國西部)	us-gov-west-1

使用 AWS ParallelCluster

主題

網路組態

AWS ParallelCluster 使用 Amazon Virtual Private Cloud（VPC）進行聯網。VPC 提供彈性且可設定的網路平台，您可以在其中部署叢集。

VPC 必須有 `DNS Resolution = yes`、 `DNS Hostnames = yes`和 DHCP選項，且該選項具有區域的正確網域名稱。預設DHCP選項集已指定所需的 AmazonProvidedDNS。如果指定多個網域名稱伺服器，請參閱 Amazon VPC使用者指南 中的DHCP選項集。

AWS ParallelCluster 支援下列高階組態：

- 一個子網，適用於主機和運算節點。

- 兩個子網，一個公有子網路中有主節點，而私有子網中有運算節點。子網路可以是新的或現有的子網路。

所有這些組態都可以搭配或不搭配公有 IP 定址操作。 AWS ParallelCluster 也可以部署 來針對所有 AWS 請求使用HTTP代理。這些組態的組合會產生許多部署案例。例如，您可以設定具有網際網路所有存取權的單一公有子網路。或者，您可以使用 AWS Direct Connect 和 HTTP代理為所有流量設定完全私有的網路。

請參閱以下架構圖，以取得其中某些案例的示意圖：

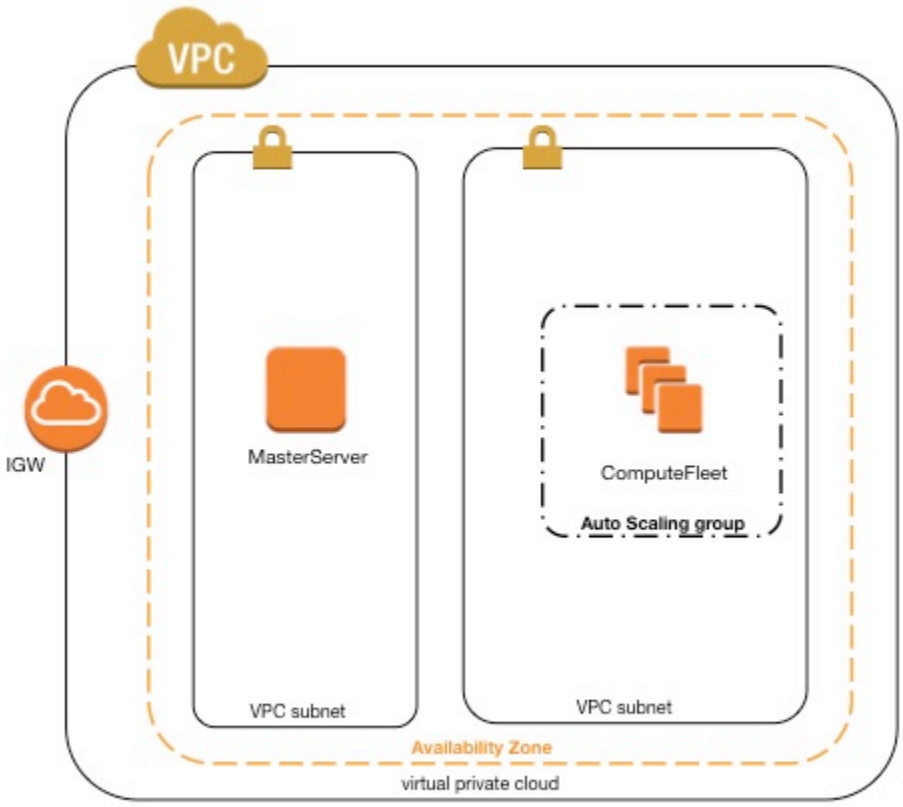
AWS ParallelCluster 在單一公有子網路中

此架構的組態需要下列設定：

```
[vpc public]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<public>
use_public_ips = true
```

此 use_public_ips 設定無法設定為 false，因為網際網路閘道要求所有執行個體都具有全域唯一的 IP 位址。如需詳細資訊，請參閱 Amazon VPC使用者指南 中的啟用網際網路存取。

AWS ParallelCluster 使用兩個子網路



若要為運算執行個體建立新的私有子網路，組態需要下列設定：

請注意，所有值僅作為範例提供。

```
[vpc public-private-new]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<public>
compute_subnet_cidr = 10.0.1.0/24
```

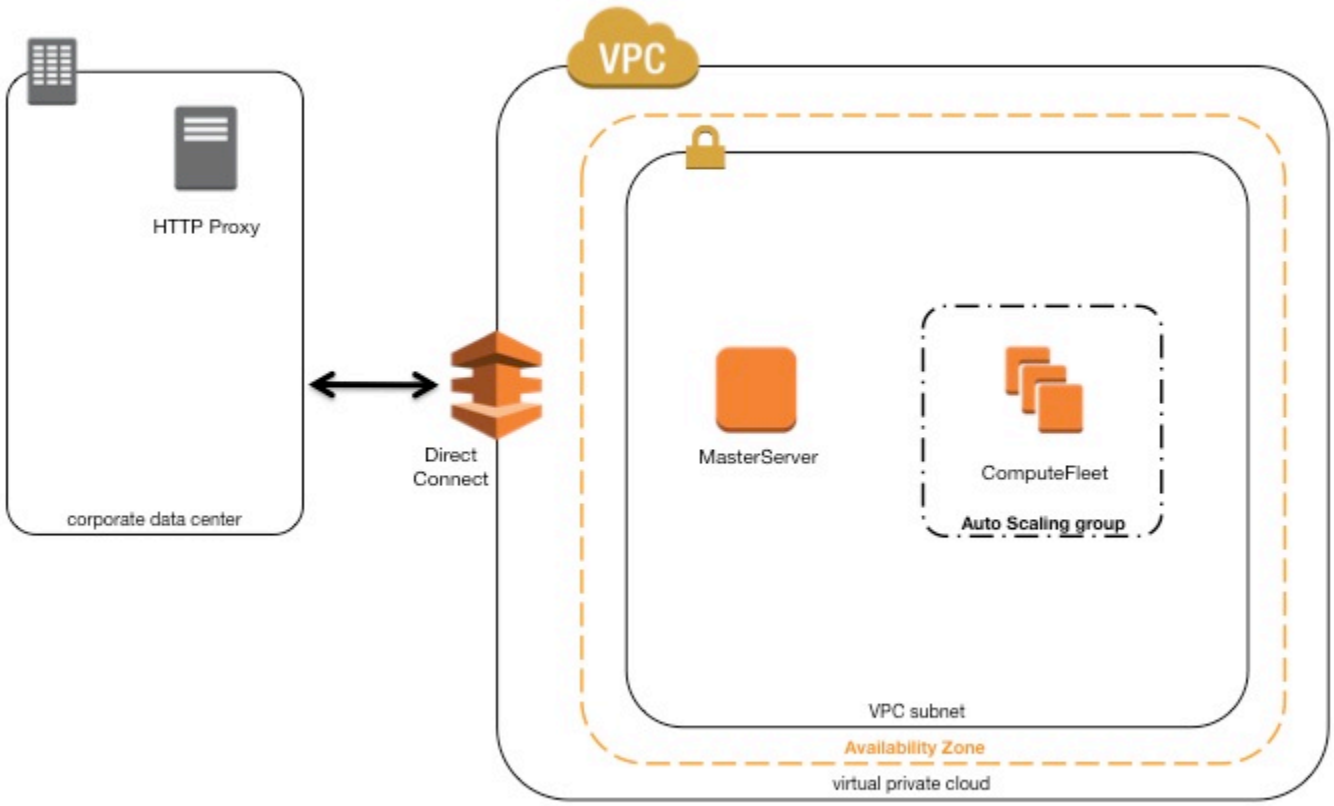
若要使用現有的私有網路，組態需要下列設定：

```
[vpc public-private-existing]
vpc_id = vpc-xxxxxx
```

```
master_subnet_id = subnet-<public>
compute_subnet_id = subnet-<private>
```

這兩種組態都需要NAT閘道或內部代理，才能為運算執行個體啟用 Web 存取。

AWS ParallelCluster 在單一私有子網路中，使用 連線 AWS Direct Connect



此架構的組態需要下列設定：

```
[cluster private-proxy]
proxy_server = http://proxy.corp.net:8080

[vpc private-proxy]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<private>
```

```
use_public_ips = false
```

當 use_public_ips 設定為 時false，VPC必須正確設定 ，才能對所有流量使用 Proxy。頭節點和運算節點都需要 Web 存取。

AWS ParallelCluster 使用**awsbatch**排程器

當您使用 awsbatch做為排程器類型時， 會 AWS ParallelCluster 建立 AWS Batch 受管運算環境。AWS Batch 環境負責管理在 中啟動的 Amazon Elastic Container Service （Amazon ECS） 容器執行個體compute_subnet。為了 AWS Batch 讓 正常運作，Amazon ECS容器執行個體需要外部網路存取權，才能與 Amazon ECS服務端點通訊。這轉換成以下案例：

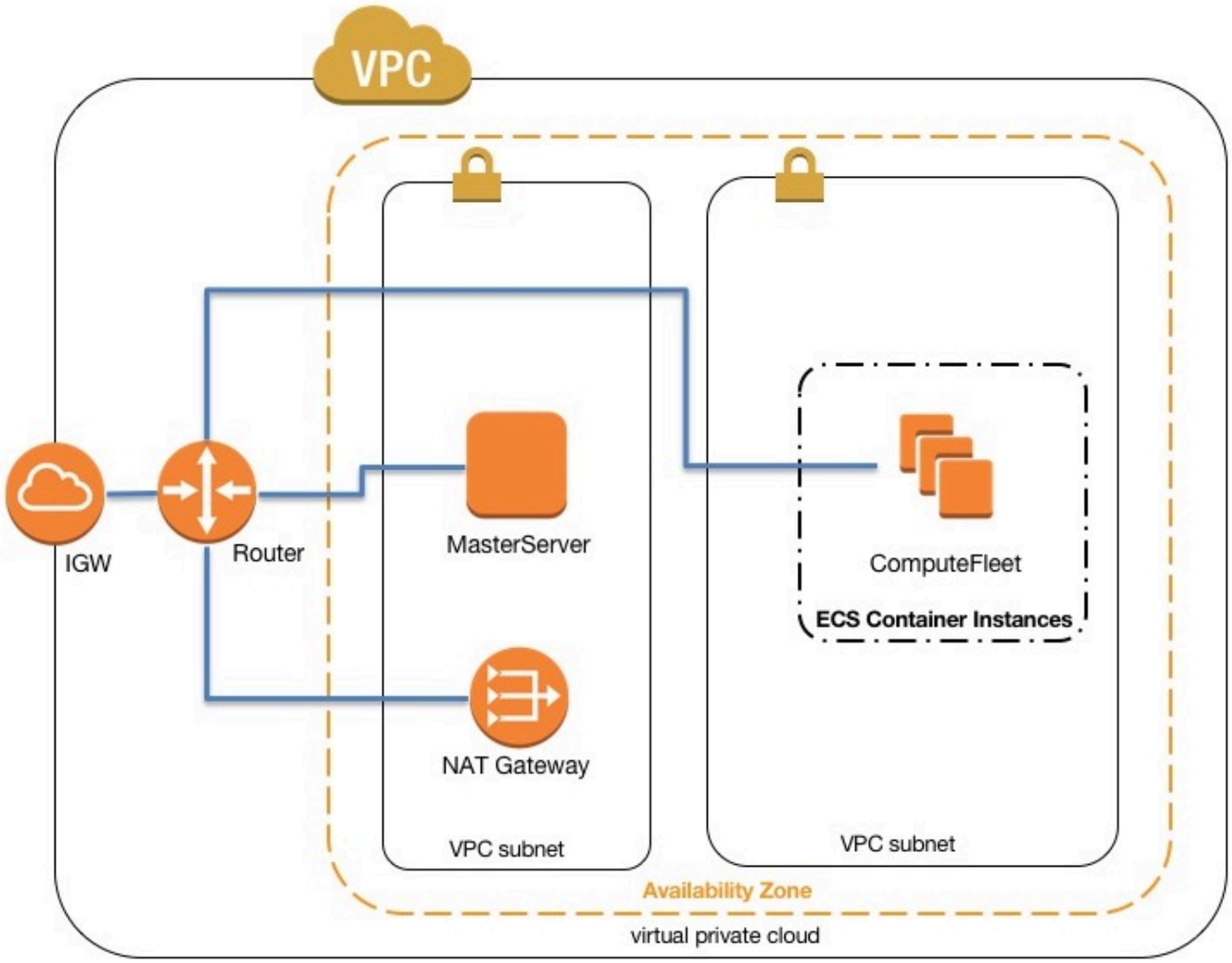
- compute_subnet 使用NAT閘道來存取網際網路。(我們建議此方法。)
- compute_subnet 中啟動的執行個體具有公有 IP 地址，而且可以透過網際網路閘道來連接網際網路。

此外，如果您對多節點平行工作感興趣 （來自AWS Batch 文件 ）：

AWS Batch 多節點平行任務使用 Amazon ECS awsvpc 網路模式，這會為您的多節點平行任務容器提供與 Amazon EC2執行個體相同的聯網屬性。每個多節點平行作業容器都會取得自己的彈性網路介面、主要私有 IP 地址和內部DNS主機名稱。網路介面與其主機運算資源在相同的 Amazon VPC子網路中建立。任何套用到您運算資源的安全群組，也會套用在它身上。

使用 Amazon ECS Task Networking 時，awsvpc網路模式不會為使用 Amazon EC2啟動類型的任務提供具有公有 IP 地址的彈性網路介面。若要存取網際網路，必須在設定為使用NAT閘道的私有子網路中啟動使用 Amazon EC2啟動類型的任務。

您必須設定NAT閘道，才能讓叢集執行多節點平行任務。



如需詳細資訊，請參閱下列主題：

- AWS Batch 受管運算環境
- AWS Batch 多節點平行任務
- Amazon ECS任務聯網與awsvpc網路模式

自訂引導操作

AWS ParallelCluster 建立叢集時， 可以在主要引導動作之前（安裝前）或之後（安裝後）執行任意程式碼。在大多數情況下，此程式碼會儲存在 Amazon Simple Storage Service（Amazon S3）中，並透過HTTPS連線存取。程式碼會以根執行，並且可以採用叢集作業系統支援的任何指令碼語言。程式碼通常位於 Bash 或 Python 中。

在啟動任何叢集部署引導動作之前，會呼叫預先安裝動作，例如設定 NAT、Amazon Elastic Block Store（Amazon EBS）或排程器。有些預先安裝動作包括修改儲存體、新增額外的使用者，以及新增套件。

安裝後動作會在叢集引導程序完成後呼叫。安裝後動作提供在執行個體視為完全設定並完成之前要執行的最後動作。部分安裝後動作包括變更排程器設定、修改儲存體和修改套件。

您可以在組態期間指定引數，藉此將引數傳遞至指令碼。為此，您可以將它們雙引號傳遞至預先安裝或安裝後動作。

如果預先安裝或安裝後動作失敗，執行個體引導也會失敗。成功會以零（0）的結束碼發出訊號。任何其他結束碼表示執行個體開機失敗。

您可以區分執行頭和運算節點。取得 `/etc/parallelcluster/cfnconfig` 檔案，並評估主機和運算節點值分別為 "`MasterServer`" 和 "`ComputeFleet`" `cfn_node_type`的環境變數。

```
#!/bin/bash

. "/etc/parallelcluster/cfnconfig"

case "${cfn_node_type}" in
    MasterServer)
        echo "I am the head node" >> /tmp/head.txt
    ;;
    ComputeFleet)
        echo "I am a compute node" >> /tmp/compute.txt
    ;;
    *)
    ;;
esac
```

組態

以下組態設定用來定義安裝前和安裝後動作和引數。

```
# URL to a preinstall script. This is run before any of the boot_as_* scripts are run
# (no default)
pre_install = https://<bucket-name>.s3.amazonaws.com/my-pre-install-script.sh
# Arguments to be passed to preinstall script
# (no default)
pre_install_args = argument-1 argument-2
# URL to a postinstall script. This is run after any of the boot_as_* scripts are run
```

```
# (no default)
post_install = https://<bucket-name>.s3.amazonaws.com/my-post-install-script.sh
# Arguments to be passed to postinstall script
# (no default)
post_install_args = argument-3 argument-4
```

引數

前兩個引數 ($0 和 $1) 保留給指令碼名稱和 url。

```
$0 => the script name
$1 => s3 url
$n => args set by pre/post_install_args
```

範例

以下步驟建立簡單的安裝後指令碼，在叢集安裝 R 套件。

1. 建立指令碼。

```
#!/bin/bash

echo "post-install script has $# arguments"
for arg in "$@"
do
    echo "arg: ${arg}"
done

yum -y install "${@:2}"
```

2. 將具有正確許可的指令碼上傳至 Amazon S3。如果公有讀取許可不適合您，請使用 s3_read_resource或 s3_read_write_resource 參數授予存取權。如需詳細資訊，請參閱使用 Amazon S3。

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

> **Important**
>
> 如果在 Windows 上編輯指令碼，則必須在將指令碼上傳至 Amazon S3 CRLF 之前，將行末尾從 變更為 LF。

3. 更新 AWS ParallelCluster 組態以包含新的安裝後動作。

```
[cluster default]
...
post_install = https://<bucket-name>.s3.amazonaws.com/myscript.sh
post_install_args = 'R curl wget'
```

如果儲存貯體沒有公有讀取許可，請使用 s3作為URL通訊協定。

```
[cluster default]
...
post_install = s3://<bucket-name>/myscript.sh
post_install_args = 'R curl wget'
```

4. 啟動叢集。

```
$ pcluster create mycluster
```

5. 驗證輸出。

```
$ less /var/log/cfn-init.log
2019-04-11 10:43:54,588 [DEBUG] Command runpostinstall output: post-install script
 has 4 arguments
arg: s3://<bucket-name>/test.sh
arg: R
arg: curl
arg: wget
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

使用 Amazon S3

若要提供存取 Amazon S3 儲存貯體的叢集資源許可，請在 ARNs中指定儲存貯體s3_read_resource，並在 AWS ParallelCluster 組態中指定s3_read_write_resource參數。如需使用 控制存取權的詳細資訊 AWS ParallelCluster，請參閱 AWS Identity and Access Management 中的 角色 AWS ParallelCluster。

```
# Specify Amazon S3 resource which AWS ParallelCluster nodes will be granted read-only
 access
# (no default)
s3_read_resource = arn:aws:s3:::my_corporate_bucket*
# Specify Amazon S3 resource which AWS ParallelCluster nodes will be granted read-write
 access
# (no default)
s3_read_write_resource = arn:aws:s3:::my_corporate_bucket/*
```

兩個參數都接受 *或有效的 Amazon S3 ARN。如需指定 Amazon S3 的相關資訊ARNs，請參閱 中的 Amazon S3 ARN 格式AWS 一般參考。

範例

下列範例可讓您讀取 Amazon S3 儲存貯體 my_corporate_bucket 中的任何物件。

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket/*
```

以下範例可讓您存取儲存貯體，但不讓您讀取儲存貯體中的項目。

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket
```

此最後一個範例可讓您讀取儲存貯體和儲存貯體中項目的存取權。

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket*
```

使用 競價型執行個體

AWS ParallelCluster 如果叢集組態已設定 `cluster_type` = Spot， 會使用 Spot 執行個體。Spot 執行個體比隨需執行個體更具成本效益，但可能會中斷。中斷的影響會因使用的特定排程器而異。利用 Spot 執行個體中斷通知，在 Amazon EC2 必須停止或終止您的 Spot 執行個體之前，提供兩分鐘的警告，可能會有所幫助。如需詳細資訊，請參閱 Amazon EC2 使用者指南 中的 Spot 執行個體中斷。下列章節說明可能中斷 Spot 執行個體的三種案例。

> **Note**
>
> 使用 Spot 執行個體需要 帳戶中存在AWSServiceRoleForEC2Spot服務連結角色。若要使用在帳戶中建立此角色 AWS CLI，請執行下列命令：

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

如需詳細資訊，請參閱 Amazon EC2使用者指南 中的 Spot 執行個體請求的服務連結角色。

案例 1：沒有執行中任務的 Spot 執行個體遭到中斷

發生此中斷時，如果排程器佇列有需要其他執行個體的待處理任務，或作用中執行個體的數量低於`initial_queue_size`設定，則 AWS ParallelCluster 會嘗試取代執行個體。如果 AWS ParallelCluster 無法佈建新的執行個體，則會定期重複對新執行個體的請求。

案例 2：執行單一節點任務的 Spot 執行個體遭到中斷

此類中斷的行為取決於所使用的排程器。

Slurm

任務失敗，狀態碼為 `NODE_FAIL`，且任務會重新排入佇列 （ 除非在提交任務時`--no-requeue`指定 ）。如果節點是靜態節點，則會予以取代。如果節點是動態節點，則會終止節點並重設。如需 的詳細資訊`sbatch`，包括 `--no-requeue` 參數，請參閱 sbatch 在 Slurm 文件 中。

> **Note**
>
> 此行為在 2.9.0 AWS ParallelCluster 版中已變更。較早版本以 狀態碼 終止任務，`NODE_FAIL`且節點已從排程器佇列中移除。

SGE

> **Note**
>
> 這僅適用於 AWS ParallelCluster 2.11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

任務已終止。如果任務已啟用重新執行旗標 (使用 `qsub -r yes` 或 `qalter -r yes`)，或佇列的 `rerun` 組態設定為 `TRUE`，則該任務會重新排程。運算執行個體會從排程器佇列中移除。此行為來自下列SGE組態參數：

- `reschedule_unknown 00:00:30`
- `ENABLE_FORCED_QDEL_IF_UNKNOWN`
- `ENABLE_RESCHEDULE_KILL=1`

Torque

> **Note**
>
> 這僅適用於 AWS ParallelCluster 2.11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

任務會從系統中移除，並從排程器中移除節點。任務不會重新執行。如果執行個體在中斷時執行多個任務，則在節點移除期間，扭矩可能會逾時。`sqswatcher` 日誌檔案中可能會顯示錯誤。這不會影響擴展邏輯，且後續重試會執行適當的清除。

案例 3：執行多節點任務的 Spot 執行個體遭到中斷

此類中斷的行為取決於所使用的排程器。

Slurm

任務失敗，狀態碼為 `NODE_FAIL`，且任務已重新排入佇列（除非提交任務時`--no-requeue`已指定）。如果節點是靜態節點，則會予以取代。如果節點是動態節點，則會終止節點並重設。執行終止任務的其他節點可能會配置到其他待定任務，或在設定`scaledown_idletime`的時間過後縮減規模。

> **Note**
>
> 此行為在 2.9.0 AWS ParallelCluster 版中已變更。較早版本以 狀態碼 終止任務，`NODE_FAIL`且節點已從排程器佇列中移除。其他正在執行已終止任務的節點，可能會在設定`scaledown_idletime`時間過後縮減。

SGE

> **Note**
>
> 這僅適用於 AWS ParallelCluster 2.11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

任務不會終止，並繼續在其餘節點上執行。運算節點會從排程器佇列中移除，但會在主機清單中顯示為孤立且無法使用的節點。

發生這種情況時，使用者必須刪除任務 (`qdel <jobid>`)。節點仍然會顯示在主機清單中（`qhost`），但這不會影響 AWS ParallelCluster。若要從清單中移除主機，請在取代執行個體後執行下列命令。

```
sudo -- bash -c 'source /etc/profile.d/sge.sh; qconf -dattr hostgroup
 hostlist <hostname> @allhosts; qconf -de <hostname>'
```

Torque

> **Note**
>
> 這僅適用於 AWS ParallelCluster 2.11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

任務會從系統中移除，並從排程器中移除節點。任務不會重新執行。如果執行個體在中斷時執行多個任務，則在節點移除期間，扭矩可能會逾時。`sqswatcher` 日誌檔案中可能會顯示錯誤。這不會影響擴展邏輯，且後續重試會執行適當的清除。

如需 Spot 執行個體的詳細資訊，請參閱 Amazon EC2使用者指南 中的 Spot 執行個體。

AWS Identity and Access Management 中的 角色 AWS ParallelCluster

AWS ParallelCluster 使用 AWS Identity and Access Management（IAM）角色讓EC2執行個體存取叢集的部署和操作 AWS 服務。根據預設，Amazon IAM的角色EC2會在建立叢集時建立。這表示建立叢集的使用者必須擁有適當層級的許可，如以下章節所述。

AWS ParallelCluster 使用多個 AWS 服務來部署和操作叢集。請參閱 章節AWS 中使用的服務 AWS ParallelCluster的完整清單。

您可以在 AWS ParallelCluster 上的文件中 GitHub追蹤範例政策的變更。

主題

叢集建立的預設設定

當您使用預設設定建立叢集時，叢集EC2會建立 Amazon 的預設IAM角色。建立叢集的使用者必須具有適當的許可層級，才能建立啟動叢集所需的所有資源。這包括為 Amazon 建立IAM角色EC2。一般而言，使用者在使用預設設定時，必須擁有AdministratorAccess受管政策的許可。如需受管政策的相關資訊，請參閱 IAM 使用者指南 中的AWS 受管政策。

使用 Amazon 的現有IAM角色 EC2

建立叢集`ec2_iam_role`時，您可以使用現有的 ，但必須先定義IAM政策和角色，才能嘗試啟動叢集。通常，您可以選擇 Amazon 的現有IAM角色EC2，將啟動叢集時授予使用者的許可降至最低。AWS ParallelCluster 範例執行個體和使用者政策 包含 AWS ParallelCluster 及其功能所需的最低許可。您必須在 中將政策和角色建立為個別政策，IAM然後將角色和政策連接到適當的資源。某些角色政策可能會變得很大，並導致配額錯誤。如需詳細資訊，請參閱對IAM政策大小問題進行故障診斷。在政策中，取代 *<REGION>*, *<AWS ACCOUNT ID>*，以及具有適當值的類似字串。

如果您的意圖是將額外的政策新增至叢集節點的預設設定，建議您使用`additional_iam_policies`設定傳遞額外的自訂IAM政策，而不是使用 `ec2_iam_role`設定。

AWS ParallelCluster 範例執行個體和使用者政策

下列範例政策包含 資源的 Amazon Resource Names （ ARNs ）。如果您在 AWS GovCloud (US) 或 AWS 中國分割區中工作，ARNs則必須變更 。具體而言，分割區必須從「arn：aws」變更為「arn：aws-us-gov」， AWS GovCloud (US) 而 AWS 中國分割區則必須從「arn：aws-cn」變更為「arn：aws-cn」。如需詳細資訊，請參閱 AWS GovCloud (US) 使用者指南 中的 AWS GovCloud (US) 區域中的 Amazon Resource Names （ ARNs ）ARNs，以及在中國 開始使用中國 AWS中的 服務。 AWS

這些政策包含 、 AWS ParallelCluster其功能和資源目前所需的最低許可。某些角色政策可能會變得很大，並導致配額錯誤。如需詳細資訊，請參閱對IAM政策大小問題進行故障診斷。

主題

ParallelClusterInstancePolicy 使用 SGE, Slurm，或 Torque

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。您可以在 和 2.11.4 之前的版本中繼續使用這些版本，但它們不符合 AWS 服務與 AWS 支援團隊未來更新或故障診斷支援的資格。

主題

ParallelClusterInstancePolicy 使用 Slurm

下列範例`ParallelClusterInstancePolicy`使用 設定 Slurm 作為排程器。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:AttachVolume",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
```

```
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstances",
                "ec2:DescribeRegions",
                "ec2:TerminateInstances",
                "ec2:DescribeLaunchTemplates",
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "EC2"
        },
        {
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:subnet/<COMPUTE SUBNET ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:network-interface/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*",
                "arn:aws:ec2:<REGION>::image/<IMAGE ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:key-pair/<KEY NAME>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:security-group/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:launch-template/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:placement-group/*"
            ],
            "Effect": "Allow",
            "Sid": "EC2RunInstances"
        },
        {
            "Action": [
                "dynamodb:ListTables"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBList"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
```

```
            ],
            "Resource": [
                "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:GetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:DeleteItem",
                "dynamodb:DescribeTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBTable"
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
            ],
            "Effect": "Allow",
            "Sid": "S3GetObj"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "IAMPassRole",
            "Condition": {
                "StringEquals": {
```

```
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::dcv-license.<REGION>/*"
            ],
            "Effect": "Allow",
            "Sid": "DcvLicense"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "GetClusterConfig"
        },
        {
            "Action": [
                "fsx:DescribeFileSystems"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "*"
```

```
            ],
            "Effect": "Allow",
            "Sid": "CWLogs"
        },
        {
            "Action": [
                "route53:ChangeResourceRecordSets"
            ],
            "Resource": [
                "arn:aws:route53:::hostedzone/*"
            ],
            "Effect": "Allow",
            "Sid": "Route53"
        }
    ]
}
```

ParallelClusterInstancePolicy 使用 SGE 或 Torque

下列範例ParallelClusterInstancePolicy使用 設定 SGE 或 Torque 作為排程器。

> **Note**
>
> 此政策僅適用於 AWS ParallelCluster 2.11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:AttachVolume",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstances",
                "ec2:DescribeRegions",
                "ec2:TerminateInstances",
                "ec2:DescribeLaunchTemplates",
                "ec2:CreateTags"
```

```
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "EC2"
        },
        {
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:subnet/<COMPUTE SUBNET ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:network-interface/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*",
                "arn:aws:ec2:<REGION>::image/<IMAGE ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:key-pair/<KEY NAME>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:security-group/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:launch-template/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:placement-group/*"
            ],
            "Effect": "Allow",
            "Sid": "EC2RunInstances"
        },
        {
            "Action": [
                "dynamodb:ListTables"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBList"
        },
        {
            "Action": [
                "sqs:SendMessage",
                "sqs:ReceiveMessage",
                "sqs:ChangeMessageVisibility",
                "sqs:DeleteMessage",
                "sqs:GetQueueUrl"
            ],
            "Resource": [
                "arn:aws:sqs:<REGION>:<AWS ACCOUNT ID>:parallelcluster-*"
            ],
```

```
            "Effect": "Allow",
            "Sid": "SQSQueue"
        },
        {
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DescribeTags",
                "autoscaling:SetInstanceHealth"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "Autoscaling"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
            ],
            "Resource": [
                "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:GetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:DeleteItem",
                "dynamodb:DescribeTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*"
            ],
            "Effect": "Allow",
```

```
            "Sid": "DynamoDBTable"
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
            ],
            "Effect": "Allow",
            "Sid": "S3GetObj"
        },
        {
            "Action": [
                "sqs:ListQueues"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "SQSList"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "IAMPassRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::dcv-license.<REGION>/*"
            ],
            "Effect": "Allow",
            "Sid": "DcvLicense"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "GetClusterConfig"
        },
        {
            "Action": [
                "fsx:DescribeFileSystems"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "CWLogs"
        },
        {
            "Action": [
                "route53:ChangeResourceRecordSets"
            ],
            "Resource": [
                "arn:aws:route53:::hostedzone/*"
            ],
```

```
            "Effect": "Allow",
            "Sid": "Route53"
        }
    ]
}
```

ParallelClusterInstancePolicy 使用 awsbatch

下列範例會設定ParallelClusterInstancePolicy使用 awsbatch作為排程器。您必須包含指派給 AWS Batch AWS CloudFormation 巢狀堆疊中BatchUserRole定義的 的相同政策。BatchUserRole ARN 以堆疊輸出的形式提供。在此範例中，「*<RESOURCES S3 BUCKET>*」是 cluster_resource_bucket 設定的值；如果cluster_resource_bucket未指定 ，則「*<RESOURCES S3 BUCKET>*」是「parallelcluster-*」。下列範例是必要許可的概觀：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "batch:RegisterJobDefinition",
                "logs:GetLogEvents"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "batch:SubmitJob",
                "cloudformation:DescribeStacks",
                "ecs:ListContainerInstances",
                "ecs:DescribeContainerInstances",
                "logs:FilterLogEvents",
                "s3:PutObject",
                "s3:Get*",
                "s3:DeleteObject",
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-
definition/<AWS_BATCH_STACK - JOB_DEFINITION_SERIAL_NAME>:1",
```

```
                "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-
definition/<AWS_BATCH_STACK - JOB_DEFINITION_MNP_NAME>*",
                "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/<AWS_BATCH_STACK -
 JOB_QUEUE_NAME>",
                "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/<STACK NAME>/
*",
                "arn:aws:s3:::<RESOURCES S3 BUCKET>/batch/*",
                "arn:aws:iam::<AWS ACCOUNT ID>:role/<AWS_BATCH_STACK - JOB_ROLE>",
                "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/<ECS COMPUTE
 ENVIRONMENT>",
                "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/*",
                "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-
stream:*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "batch:DescribeJobQueues",
                "batch:TerminateJob",
                "batch:DescribeJobs",
                "batch:CancelJob",
                "batch:DescribeJobDefinitions",
                "batch:ListJobs",
                "batch:DescribeComputeEnvironments"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "ec2:DescribeInstances",
                "ec2:AttachVolume",
                "ec2:DescribeVolumes",
                "ec2:DescribeInstanceAttribute"
```

```
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2"
        },
        {
            "Action": [
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "fsx:DescribeFileSystems"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:TagResource",
                "logs:UntagResource",
                "logs:CreateLogStream"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "CWLogs"
        }
    ]
}
```

`ParallelClusterUserPolicy` 使用 Slurm

下列範例使用 設定 ParallelClusterUserPolicy，Slurm 作為排程器。在此範例中，「*<RESOURCES S3 BUCKET>*」是 cluster_resource_bucket 設定的值；如果cluster_resource_bucket未指定 ，則「*<RESOURCES S3 BUCKET>*」是「parallelcluster-*」。

> **Note**
>
> 如果您使用自訂角色 ec2_iam_role = *<role_name>*，則必須變更IAM資源，以包含該角色的名稱：
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"`
> 至：
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/<role_name>"`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeKeyPairs",
                "ec2:DescribeRegions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribePlacementGroups",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeAddresses",
                "ec2:CreateTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*",
```

```
            "Effect": "Allow",
            "Sid": "EC2Describe"
        },
        {
            "Action": [
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DescribeNatGateways",
                "ec2:CreateNatGateway",
                "ec2:DescribeInternetGateways",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DescribeRouteTables",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:CreateSubnet",
                "ec2:ModifySubnetAttribute"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "NetworkingEasyConfig"
        },
        {
            "Action": [
                "ec2:CreateVolume",
                "ec2:RunInstances",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AttachNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:ModifyVolumeAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteVolume",
                "ec2:TerminateInstances",
                "ec2:DeleteSecurityGroup",
                "ec2:DisassociateAddress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:ReleaseAddress",
```

```
                "ec2:CreatePlacementGroup",
                "ec2:DeletePlacementGroup"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Modify"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "ScalingModify"
        },
        {
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:ListTagsOfResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBDescribe"
        },
        {
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:TagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBModify"
        },
        {
```

```
            "Action": [
                "route53:ChangeResourceRecordSets",
                "route53:ChangeTagsForResource",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetChange",
                "route53:GetHostedZone",
                "route53:ListResourceRecordSets",
                "route53:ListQueryLoggingConfigs"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Route53HostedZones"
        },
        {
            "Action": [
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:ListStacks",
                "cloudformation:GetTemplate"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudFormationDescribe"
        },
        {
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:UpdateStack"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CloudFormationModify"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
```

```
            "Effect": "Allow",
            "Sid": "S3ResourcesBucket"
        },
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster*"
            ],
            "Effect": "Allow",
            "Sid": "S3ParallelClusterReadOnly"
        },
        {
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3Delete"
        },
        {
            "Action": [
                "iam:PassRole",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
                "iam:TagRole",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": [
                "arn:aws:iam::<AWS ACCOUNT ID>:role/<PARALLELCLUSTER EC2 ROLE NAME>",
                "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "IAMModify"
        },
        {
          "Condition": {
```

```
            "StringEquals": {
                "iam:AWSServiceName": [
                    "fsx.amazonaws.com",
                    "s3.data-source.lustre.fsx.amazonaws.com"
                ]
            }
        },
        "Action": [
            "iam:CreateServiceLinkedRole"
        ],
        "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/*",
        "Effect": "Allow",
        "Sid": "IAMServiceLinkedRole"
    },
    {
        "Action": [
            "iam:CreateInstanceProfile",
            "iam:DeleteInstanceProfile"
        ],
        "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
        "Effect": "Allow",
        "Sid": "IAMCreateInstanceProfile"
    },
    {
        "Action": [
            "iam:AddRoleToInstanceProfile",
            "iam:RemoveRoleFromInstanceProfile",
            "iam:GetRolePolicy",
            "iam:GetPolicy",
            "iam:AttachRolePolicy",
            "iam:DetachRolePolicy",
            "iam:PutRolePolicy",
            "iam:DeleteRolePolicy"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "IAMInstanceProfile"
    },
    {
        "Action": [
            "elasticfilesystem:DescribeMountTargets",
            "elasticfilesystem:DescribeMountTargetSecurityGroups",
            "ec2:DescribeNetworkInterfaceAttribute"
        ],
```

```
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFSDescribe"
        },
        {
            "Action": [
                "ssm:GetParametersByPath"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SSMDescribe"
        },
        {
            "Action": [
                "fsx:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "elasticfilesystem:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFS"
        },
        {
            "Action": [
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup",
                "logs:TagResource",
                "logs:UntagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudWatchLogs"
        },
        {
            "Action": [
                "lambda:CreateFunction",
```

```
                "lambda:DeleteFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:AddPermission",
                "lambda:RemovePermission",
                "lambda:TagResource",
                "lambda:ListTags",
                "lambda:UntagResource"
            ],
            "Resource": [
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "Lambda"
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
              "cloudwatch:PutDashboard",
              "cloudwatch:ListDashboards",
              "cloudwatch:DeleteDashboards",
              "cloudwatch:GetDashboard"
            ],
            "Resource": "*"
        }
    ]
}
```

ParallelClusterUserPolicy 使用 SGE 或 Torque

> **Note**
>
> 本節僅適用於 2 AWS ParallelCluster .11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

下列範例使用 設定 `ParallelClusterUserPolicy`，SGE 或 Torque 作為排程器。在此範例中，「*<RESOURCES S3 BUCKET>*」是 cluster_resource_bucket 設定的值；如

果cluster_resource_bucket未指定 ，則「*<RESOURCES S3 BUCKET>*」是「parallelcluster-*」。

> **Note**
>
> 如果您使用自訂角色 ec2_iam_role = *<role_name>*，則必須變更IAM資源，以包含該角色的名稱：
>
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"`
>
> 至：
>
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/<role_name>"`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeKeyPairs",
                "ec2:DescribeRegions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribePlacementGroups",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeAddresses",
                "ec2:CreateTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Describe"
        },
        {
            "Action": [
```

```
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DescribeNatGateways",
                "ec2:CreateNatGateway",
                "ec2:DescribeInternetGateways",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DescribeRouteTables",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:CreateSubnet",
                "ec2:ModifySubnetAttribute"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "NetworkingEasyConfig"
        },
        {
            "Action": [
                "ec2:CreateVolume",
                "ec2:RunInstances",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AttachNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:ModifyVolumeAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteVolume",
                "ec2:TerminateInstances",
                "ec2:DeleteSecurityGroup",
                "ec2:DisassociateAddress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:ReleaseAddress",
                "ec2:CreatePlacementGroup",
                "ec2:DeletePlacementGroup"
            ],
            "Resource": "*",
            "Effect": "Allow",
```

```
            "Sid": "EC2Modify"
        },
        {
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AutoScalingDescribe"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:PutScalingPolicy",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeletePolicy",
                "autoscaling:DisableMetricsCollection",
                "autoscaling:EnableMetricsCollection"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AutoScalingModify"
        },
        {
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:ListTagsOfResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBDescribe"
        },
        {
            "Action": [
```

```
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:TagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBModify"
        },
        {
            "Action": [
                "sqs:GetQueueAttributes"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SQSDescribe"
        },
        {
            "Action": [
                "sqs:CreateQueue",
                "sqs:SetQueueAttributes",
                "sqs:DeleteQueue",
                "sqs:TagQueue"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SQSModify"
        },
        {
            "Action": [
                "sns:ListTopics",
                "sns:GetTopicAttributes"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SNSDescribe"
        },
        {
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Unsubscribe",
```

```
                "sns:DeleteTopic"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SNSModify"
        },
        {
            "Action": [
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:ListStacks",
                "cloudformation:GetTemplate"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudFormationDescribe"
        },
        {
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:UpdateStack"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CloudFormationModify"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3ResourcesBucket"
        },
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster*"
            ],
            "Effect": "Allow",
            "Sid": "S3ParallelClusterReadOnly"
        },
        {
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3Delete"
        },
        {
            "Action": [
                "iam:PassRole",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
                "iam:TagRole",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": [
                "arn:aws:iam::<AWS ACCOUNT ID>:role/<PARALLELCLUSTER EC2 ROLE NAME>",
                "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "IAMModify"
        },
        {
          "Condition": {
              "StringEquals": {
                  "iam:AWSServiceName": [
                      "fsx.amazonaws.com",
                      "s3.data-source.lustre.fsx.amazonaws.com"
                  ]
              }
          },
          "Action": [
```

```
            "iam:CreateServiceLinkedRole"
        ],
        "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/*",
        "Effect": "Allow",
        "Sid": "IAMServiceLinkedRole"
    },
    {
        "Action": [
            "iam:CreateInstanceProfile",
            "iam:DeleteInstanceProfile"
        ],
        "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
        "Effect": "Allow",
        "Sid": "IAMCreateInstanceProfile"
    },
    {
        "Action": [
            "iam:AddRoleToInstanceProfile",
            "iam:RemoveRoleFromInstanceProfile",
            "iam:GetRolePolicy",
            "iam:GetPolicy",
            "iam:AttachRolePolicy",
            "iam:DetachRolePolicy",
            "iam:PutRolePolicy",
            "iam:DeleteRolePolicy"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "IAMInstanceProfile"
    },
    {
        "Action": [
            "elasticfilesystem:DescribeMountTargets",
            "elasticfilesystem:DescribeMountTargetSecurityGroups",
            "ec2:DescribeNetworkInterfaceAttribute"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "EFSDescribe"
    },
    {
        "Action": [
            "ssm:GetParametersByPath"
        ],
```

```
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SSMDescribe"
        },
        {
            "Action": [
                "fsx:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "elasticfilesystem:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFS"
        },
        {
            "Action": [
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup",
                "logs:TagResource",
                "logs:UntagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudWatchLogs"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:AddPermission",
                "lambda:RemovePermission",
                "lambda:TagResource",
                "lambda:ListTags",
```

```
                "lambda:UntagResource"
            ],
            "Resource": [
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "Lambda"
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
              "cloudwatch:PutDashboard",
              "cloudwatch:ListDashboards",
              "cloudwatch:DeleteDashboards",
              "cloudwatch:GetDashboard"
            ],
            "Resource": "*"
        }
    ]
}
```

ParallelClusterUserPolicy 使用 awsbatch

下列範例ParallelClusterUserPolicy使用 awsbatch作為排程器。在此範例中，「*<RESOURCES S3 BUCKET>*」是 cluster_resource_bucket 設定的值；如果cluster_resource_bucket未指定 ，則「*<RESOURCES S3 BUCKET>*」是「parallelcluster-*」。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeKeyPairs",
                "ec2:DescribeRegions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribePlacementGroups",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
```

```
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeAddresses",
                "ec2:CreateTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Describe"
        },
        {
            "Action": [
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2LaunchTemplate"
        },
        {
            "Action": [
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DescribeNatGateways",
                "ec2:CreateNatGateway",
                "ec2:DescribeInternetGateways",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DescribeRouteTables",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:CreateSubnet",
                "ec2:ModifySubnetAttribute"
            ],
```

```
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "NetworkingEasyConfig"
        },
        {
            "Action": [
                "ec2:CreateVolume",
                "ec2:RunInstances",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AttachNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:ModifyVolumeAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteVolume",
                "ec2:TerminateInstances",
                "ec2:DeleteSecurityGroup",
                "ec2:DisassociateAddress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:ReleaseAddress",
                "ec2:CreatePlacementGroup",
                "ec2:DeletePlacementGroup"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Modify"
        },
        {
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:TagResource"
            ],
            "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/
parallelcluster-*",
```

```
            "Effect": "Allow",
            "Sid": "DynamoDB"
        },
        {
            "Action": [
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:ListStacks",
                "cloudformation:GetTemplate",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:UpdateStack"
            ],
            "Resource": "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*",
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "route53:ChangeResourceRecordSets",
                "route53:ChangeTagsForResource",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetChange",
                "route53:GetHostedZone",
                "route53:ListResourceRecordSets"
            ],
            "Resource": "arn:aws:route53:::hostedzone/*",
            "Effect": "Allow",
            "Sid": "Route53HostedZones"
        },
        {
            "Action": [
                "sqs:GetQueueAttributes",
                "sqs:CreateQueue",
                "sqs:SetQueueAttributes",
                "sqs:DeleteQueue",
                "sqs:TagQueue"
            ],
            "Resource": "*",
            "Effect": "Allow",
```

```
            "Sid": "SQS"
        },
        {
            "Action": [
                "sqs:SendMessage",
                "sqs:ReceiveMessage",
                "sqs:ChangeMessageVisibility",
                "sqs:DeleteMessage",
                "sqs:GetQueueUrl"
            ],
            "Resource": "arn:aws:sqs:<REGION>:<AWS ACCOUNT ID>:parallelcluster-*",
            "Effect": "Allow",
            "Sid": "SQSQueue"
        },
        {
            "Action": [
                "sns:ListTopics",
                "sns:GetTopicAttributes",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sns:DeleteTopic"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SNS"
        },
        {
            "Action": [
                "iam:PassRole",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
                "iam:TagRole",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": [
                "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*",
                "arn:aws:iam::<AWS ACCOUNT ID>:role/<PARALLELCLUSTER EC2 ROLE NAME>"
            ],
            "Effect": "Allow",
            "Sid": "IAMRole"
        },
        {
```

```
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
            "Effect": "Allow",
            "Sid": "IAMInstanceProfile"
        },
        {
            "Action": [
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:GetRolePolicy",
                "iam:PutRolePolicy",
                "iam:DeleteRolePolicy",
                "iam:GetPolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "IAM"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3ResourcesBucket"
        },
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
            ],
            "Effect": "Allow",
```

```
            "Sid": "S3ParallelClusterReadOnly"
        },
        {
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3Delete"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:InvokeFunction",
                "lambda:AddPermission",
                "lambda:RemovePermission",
                "lambda:TagResource",
                "lambda:ListTags",
                "lambda:UntagResource"
            ],
            "Resource": [
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "Lambda"
        },
        {
            "Action": [
                "logs:*"
            ],
            "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:*",
            "Effect": "Allow",
            "Sid": "Logs"
        },
        {
            "Action": [
```

```
                "codebuild:*"
            ],
            "Resource": "arn:aws:codebuild:<REGION>:<AWS ACCOUNT ID>:project/
parallelcluster-*",
            "Effect": "Allow",
            "Sid": "CodeBuild"
        },
        {
            "Action": [
                "ecr:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "ECR"
        },
        {
            "Action": [
                "batch:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Batch"
        },
        {
            "Action": [
                "events:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "AmazonCloudWatchEvents"
        },
        {
            "Action": [
                "ecs:DescribeContainerInstances",
                "ecs:ListContainerInstances"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "ECS"
        },
        {
            "Action": [
                "elasticfilesystem:CreateFileSystem",
                "elasticfilesystem:CreateMountTarget",
```

```
                "elasticfilesystem:DeleteFileSystem",
                "elasticfilesystem:DeleteMountTarget",
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:DescribeMountTargets"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFS"
        },
        {
            "Action": [
                "fsx:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
              "cloudwatch:PutDashboard",
              "cloudwatch:ListDashboards",
              "cloudwatch:DeleteDashboards",
              "cloudwatch:GetDashboard"
            ],
            "Resource": "*"
        }
    ]
}
```

`ParallelClusterLambdaPolicy` 使用 SGE, Slurm，或 Torque

下列範例使用 設定 `ParallelClusterLambdaPolicy`，SGE, Slurm，或 Torque 作為排程器。

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

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

```
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*",
      "Effect": "Allow",
      "Sid": "CloudWatchLogsPolicy"
    },
    {
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Effect": "Allow",
      "Sid": "S3BucketPolicy"
    },
    {
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "DescribeInstances"
    },
    {
      "Action": [
        "ec2:TerminateInstances"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "FleetTerminatePolicy"
    },
    {
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem"
```

```
      ],
      "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
      "Effect": "Allow",
      "Sid": "DynamoDBTable"
    },
    {
      "Action": [
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": [
        "arn:aws:route53:::hostedzone/*"
      ],
      "Effect": "Allow",
      "Sid": "Route53DeletePolicy"
    }
  ]
}
```

ParallelClusterLambdaPolicy 使用 awsbatch

下列範例設定ParallelClusterLambdaPolicy使用 awsbatch作為排程器。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:*:*:*",
      "Sid": "CloudWatchLogsPolicy"
    },
    {
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:ListImages"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "ECRPolicy"
```

```
    },
    {
      "Action": [
        "codebuild:BatchGetBuilds",
        "codebuild:StartBuild"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "CodeBuildPolicy"
    },
    {
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "S3BucketPolicy"
    }
  ]
}
```

`ParallelClusterUserPolicy` 適用於使用者

下列範例`ParallelClusterUserPolicy`為不需要建立或更新叢集的使用者設定 。支援下列命令。

- pcluster dcv
- pcluster instances
- pcluster list
- pcluster ssh
- pcluster start
- pcluster status
- pcluster stop
- pcluster version

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
             "Sid": "MinimumModify",
             "Action": [
                "autoscaling:UpdateAutoScalingGroup",
                "batch:UpdateComputeEnvironment",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResources",
                "cloudformation:GetTemplate",
                "dynamodb:GetItem",
                "dynamodb:PutItem"
             ],
             "Effect": "Allow",
             "Resource": [
                 "arn:aws:autoscaling:<REGION>:<AWS ACCOUNT
 ID>:autoScalingGroup:*:autoScalingGroupName/parallelcluster-*",
                 "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:compute-environment/*",
                 "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/<CLUSTERNAME>/
*",
                 "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/<CLUSTERNAME>"
             ]
        },
        {
           "Sid": "Describe",
           "Action": [
                "cloudformation:DescribeStacks",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus"
           ],
           "Effect": "Allow",
           "Resource": "*"
        }
    ]
}
```

支援的排程器 AWS ParallelCluster

AWS ParallelCluster 支援數個排程器，請使用 設定進行scheduler設定。

> **Note**
>
> 從 2.11.5 版開始，AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。您可以在 和 2.11.4 之前的版本中繼續使用這些版本，但它們不符合 AWS 服務與 AWS 支援團隊未來更新或故障診斷支援的資格。

主題

Son of Grid Engine (`sge`)

> **Note**
>
> 從 2.11.5 版開始，AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。您可以在 和 2.11.4 之前的版本中繼續使用這些版本，但它們不符合 AWS 服務與 AWS 支援團隊未來更新或故障診斷支援的資格。

AWS ParallelCluster 2.11.4 版及更早版本使用 Son of Grid Engine 8.1.9。

Slurm Workload Manager (`slurm`)

AWS ParallelCluster 2.11.9 版使用 Slurm 20.11.9。如需有關 的資訊 Slurm，請參閱https://slurm.schedmd.com/。如需下載，請參閱 https://github.com/SchedMD/slurm/tags。如需來源碼，請參閱「https://github.com/SchedMD/slurm」。

> **Important**
>
> AWS ParallelCluster 已使用 進行測試 Slurm 預設提供的組態參數。您對這些項目所做的任何變更 Slurm 組態參數需自負風險。僅以最佳努力為基礎支援它們。

AWS ParallelCluster version（s）	支援 Slurm version
2.11.7、2.11.8、2.11.9	20.11.9
2.11.4 至 2.11.6	20.11.8
2.11.0 至 2.11.3	20.11.7
2.10.4	20.02.7
2.9.0 至 2.10.3	20.02.4
2.6 至 2.8.1	19.05.5
2.5.0、2.5.1	19.05.3-2
2.3.1 至 2.4.1	18.08.6-2
2.3.1 之前	16.05.3-1

多重佇列模式

AWS ParallelCluster 2.9.0 版推出多個佇列模式。當 scheduler 設定為 slurm且queue_settings設定已定義時，支援多個佇列模式。此模式允許不同的執行個體類型在運算節點中共存。包含不同執行個體類型的運算資源可以視需要擴展或縮減。在佇列模式中，最多支援五（5）個佇列，每個[queue]區段最多可以參考三（3）個[compute_resource]區段。每個[queue]區段都是 中的分割區 Slurm Workload Manager。 如需詳細資訊，請參閱 Slurm 多個佇列模式的指南和 多隊列模式教程。

佇列中的每個[compute_resource]區段都必須具有不同的執行個體類型，且每個區段[compute_resource]都進一步分為靜態和動態節點。每個 的靜態節點[compute_resource]編號從 1 到 的值min_count。每個 的動態節點[compute_resource]從一（1）編號為（max_count - min_count）。例如，如果 min_count 是 2 且 max_count是 10，則 的動態節點[compute_resource]編號從一（1）到八（8）。在任何時間， 中的動態節點數目上限可以介於零（0）和 之間[compute_resource]。

會動態指派在運算機群中啟動的執行個體。為了協助管理此項目，會為每個節點產生主機名稱。主機名稱的格式如下：

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

- $QUEUE 是佇列的名稱。例如，如果區段開始[queue *queue-name*]，則「$QUEUE」為「*queue-name*”.
- $STATDYN st適用於靜態節點或dy動態節點。
- $INSTANCE_TYPE 是 的執行個體類型[compute_resource]，來自 instance_type設定。
- $NODENUM 是節點的數目。min_count靜態節點$NODENUM的值介於一（1）和 之間，動態節點的值介於一（1）和（max_count - min_count）之間。

主機名稱和完整網域名稱（FQDN）都是使用 Amazon Route 53 託管區域建立。FQDN 是 $HOSTNAME.$CLUSTERNAME.pcluster，其中 $CLUSTERNAME是用於叢集的[cluster]區段名稱。

若要將組態轉換為佇列模式，請使用 pcluster-config convert命令。它使用名為 的單一[queue]區段寫入更新的組態[queue compute]。該佇列包含名為 的單一[compute_resource]區段[compute_resource default]。[queue compute] 和 [compute_resource default]的設定已從指定的 [cluster] 區段遷移。

Slurm 多個佇列模式的指南

AWS ParallelCluster 2.9.0 版為 推出多個佇列模式和新的擴展架構 Slurm Workload Manager (Slurm).

下列各節提供使用 的一般概觀 Slurm 使用新推出的擴展架構。

概觀

新的擴展架構是以 為基礎 Slurm的 Cloud Scheduling 指南和省電外掛程式。如需省電外掛程式的詳細資訊，請參閱 Slurm 省電指南 。在新的架構中，可能提供給叢集的資源通常預先定義在 中 Slurm 組態為雲端節點。

雲端節點生命週期

在整個生命週期中，如果不是下列所有狀態，則雲端節點會輸入數項：POWER_SAVING、POWER_UP（pow_up）、 ALLOCATED （alloc） 和 POWER_DOWN（pow_dn）。在某些情況下，雲端節點可能會進入 OFFLINE 狀態。下列清單詳細說明了雲端節點生命週期中這些狀態的幾個方面。

- 狀態中的節點會在 中POWER_SAVING顯示~尾碼（例如 idle~）sinfo。在此狀態下，沒有EC2執行個體可備份節點。不過，Slurm 仍然可以將任務配置到節點。
- 轉換為 POWER_UP 狀態的節點會在 中顯示#尾碼（例如 idle#）sinfo。
- 當 Slurm 會將任務配置到處於 POWER_SAVING 狀態的節點，節點會自動轉移到 POWER_UP 狀態。否則，可以使用 scontrol update nodename=*nodename* state=power_up命令手動將節

點置於 POWER_UP 狀態。在此階段， 會ResumeProgram叫用，並EC2啟動執行個體並設定 來傳回POWER_UP節點。

- 目前可供使用的節點會在 中顯示沒有任何字尾（例如 idle）sinfo。節點設定並加入叢集後，即可執行任務。在此階段中，節點已正確設定並可供使用。一般而言，建議您使用 EC2 中的執行個體數目與可用節點數目相同。在大多數情況下，建立叢集後，靜態節點一律可用。
- 轉換至 POWER_DOWN 狀態的節點會在 中顯示%尾碼（例如 idle%）sinfo。動態節點會在 之後自動進入 POWER_DOWN 狀態scaledown_idletime。相比之下，大多數情況下的靜態節點不會關閉電源。不過，可以使用 scontrol update nodename=*nodename* state=powering_down命令手動將節點置於 POWER_DOWN 狀態。在此狀態下，與節點相關聯的執行個體會終止，而節點會在之後重設回 POWER_SAVING 狀態以供未來使用scaledown_idletime。scaledown-idletime設定會儲存至 Slurm 組態作為SuspendTimeout設定。
- 離線的節點會在 中顯示*尾碼（例如 down*）sinfo。節點離線，如果 Slurm 控制器無法聯絡節點，或者如果停用靜態節點，且備份執行個體終止。

現在請考慮下列sinfo範例中顯示的節點狀態。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa          up   infinite      4  idle~ efa-dy-c5n18xlarge-[1-4]
efa          up   infinite      1   idle efa-st-c5n18xlarge-1
gpu          up   infinite      1  idle% gpu-dy-g38xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]
ondemand     up   infinite      2   mix# ondemand-dy-c52xlarge-[1-2]
ondemand     up   infinite     18  idle~ ondemand-dy-c52xlarge-[3-10],ondemand-dy-
t2xlarge-[1-10]
spot*        up   infinite     13  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2large-[1-3]
spot*        up   infinite      2   idle spot-st-t2large-[1-2]
```

spot-st-t2large-[1-2] 和 efa-st-c5n18xlarge-1節點已設定備份執行個體，且可供使用。ondemand-dy-c52xlarge-[1-2] 節點處於 POWER_UP 狀態，應該會在幾分鐘內可用。gpu-dy-g38xlarge-1 節點處於 POWER_DOWN 狀態，且會在 scaledown_idletime（預設為 120 秒）後轉換為 POWER_SAVING 狀態。

所有其他節點都處於 POWER_SAVING 狀態，沒有EC2執行個體支援它們。

使用可用的節點

可用節點由EC2執行個體支援。根據預設，節點名稱可以直接用於執行個體 SSH（例如 ssh efa-st-c5n18xlarge-1）。執行個體的私有 IP 地址可以使用 scontrol show nodes *nodename*命

令擷取，並檢查 NodeAddr 欄位。對於無法使用的節點，NodeAddr欄位不應指向執行中的EC2執行個體。相反地，它應該與節點名稱相同。

任務狀態和提交

在大多數情況下提交的任務會立即配置到系統中的節點，或者如果配置了所有節點，則會置於待定狀態。

如果為任務配置的節點包含處於 POWER_SAVING 狀態的任何節點，任務會以 CF、 或 CONFIGURING 狀態開始。此時，任務會等待 POWER_SAVING 狀態中的節點轉換為 POWER_UP 狀態並變為可用。

為任務配置的所有節點都可用後，任務會進入 RUNNING（R） 狀態。

根據預設，所有任務都會提交至預設佇列（在 中稱為分割區 Slurm）。 這由佇列名稱後面的字*尾表示。您可以使用-p任務提交選項來選取佇列。

所有節點都具有下列功能，可用於任務提交命令：

- 執行個體類型（例如 c5.xlarge）
- 節點類型（這是 dynamic或 static。）

您可以使用 scontrol show nodes *nodename*命令並檢查AvailableFeatures清單，以查看特定節點的所有可用功能。

另一個考量因素是任務。首先考慮叢集的初始狀態，您可以透過執行 sinfo命令來檢視該狀態。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa          up   infinite      4  idle~ efa-dy-c5n18xlarge-[1-4]
efa          up   infinite      1   idle efa-st-c5n18xlarge-1
gpu          up   infinite     10  idle~ gpu-dy-g38xlarge-[1-10]
ondemand     up   infinite     20  idle~ ondemand-dy-c52xlarge-[1-10],ondemand-dy-
t2xlarge-[1-10]
spot*        up   infinite     13  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2large-[1-3]
spot*        up   infinite      2   idle spot-st-t2large-[1-2]
```

請注意， spot是預設佇列。它以*尾碼表示。

將任務提交至一個靜態節點至預設佇列（spot）。

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

將任務提交至EFA佇列的一個動態節點。

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

將任務提交至八（8）個c5.2xlarge節點，並將兩（2）個t2.xlarge節點提交至ondemand佇列。

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

將任務提交至gpu佇列的一個GPU節點。

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

現在使用 squeue命令來考慮任務的狀態。

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                12  ondemand     wrap   ubuntu CF       0:36     10 ondemand-dy-
c52xlarge-[1-8],ondemand-dy-t2xlarge-[1-2]
                13       gpu     wrap   ubuntu CF       0:05      1 gpu-dy-g38xlarge-1
                 7      spot     wrap   ubuntu  R       2:48      1 spot-st-t2large-1
                 8       efa     wrap   ubuntu  R       0:39      1 efa-dy-
c5n18xlarge-1
```

任務 7 和 8（在 spot和 efa佇列中）已在執行（）R。任務 12 和 13 仍在設定（CF），可能正在等待執行個體變成可用。

```
# Nodes states corresponds to state of running jobs
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa          up   infinite      3  idle~ efa-dy-c5n18xlarge-[2-4]
efa          up   infinite      1    mix efa-dy-c5n18xlarge-1
efa          up   infinite      1   idle efa-st-c5n18xlarge-1
gpu          up   infinite      1   mix~ gpu-dy-g38xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]
ondemand     up   infinite     10   mix# ondemand-dy-c52xlarge-[1-8],ondemand-dy-
t2xlarge-[1-2]
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[9-10],ondemand-dy-
t2xlarge-[3-10]
spot*        up   infinite     13  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2large-[1-3]
```

```
spot*        up   infinite      1    mix spot-st-t2large-1
spot*        up   infinite      1   idle spot-st-t2large-2
```

節點狀態和功能

在大多數情況下，節點狀態會 AWS ParallelCluster 依據本主題前面所述的雲端節點生命週期中的特定程序完全由 管理。

不過， AWS ParallelCluster 也會取代或終止 DOWN和 DRAINED 狀態中的運作狀態不佳節點，以及具有運作狀態不佳的備份執行個體節點。如需詳細資訊，請參閱clustermgtd。

分割區狀態

AWS ParallelCluster 支援下列分割區狀態。A Slurm 分割區是 中的佇列 AWS ParallelCluster。

- UP：表示分割區處於作用中狀態。這是分割區的預設狀態。在此狀態下，分割區中的所有節點都處於作用中狀態且可供使用。
- INACTIVE：表示分割區處於非作用中狀態。在此狀態下，停用分割區的所有執行個體備份節點都會終止。不會為非作用中分割區中的節點啟動新的執行個體。

pcluster 啟動和停止

執行 pcluster stop 時，所有分割區都會置於 INACTIVE 狀態，而 AWS ParallelCluster 程序會將分割區保持在 INACTIVE 狀態。

pcluster start 執行時，所有分割區一開始都會置於 UP 狀態。不過， AWS ParallelCluster 處理程序不會將分割區保持在 UP 狀態。您需要手動變更分割區狀態。所有靜態節點都會在幾分鐘後可用。請注意，將分割區設定為 UP不會啟動任何動態容量。如果 initial_count 大於 max_count，則當分割區狀態變更為 UP 狀態時， initial_count 可能無法滿足。

pcluster start 和 pcluster stop 執行時，您可以執行 pcluster status命令並檢查 ，藉此檢查叢集的狀態ComputeFleetStatus。下列列出可能的狀態：

- STOP_REQUESTED：pcluster stop請求會傳送至叢集。
- STOPPING：pcluster程序目前正在停止叢集。
- STOPPED：pcluster程序已完成停止程序、所有分割區都處於 INACTIVE 狀態，且所有運算執行個體都會終止。
- START_REQUESTED：pcluster start請求會傳送至叢集。

- STARTING：pcluster程序目前正在啟動叢集
- RUNNING：pcluster程序已完成啟動程序，所有分割區都處於 UP 狀態，且靜態節點將在幾分鐘後可用。

對佇列的手動控制

在某些情況下，您可能想要對節點或佇列（在 中稱為分割區 Slurm）。您可以透過下列常見程序來管理叢集中的節點。

- 啟動處於 POWER_SAVING 狀態的動態節點：執行scontrol update nodename=*nodename* state=power_up命令或提交請求特定數量節點的預留位置sleep 1任務，並依賴 Slurm 以啟動所需的節點數量。
- 在 之前關閉動態節點的電源scaledown_idletime：DOWN使用 scontrol update nodename=*nodename* state=down command. AWS ParallelCluster automatic 將動態節點設定為 。 會自動終止並重設下降的動態節點。一般而言，我們不建議您POWER_DOWN直接使用 scontrol update nodename=*nodename* state=power_down命令將節點設定為 。這是因為 AWS ParallelCluster 會自動處理關機程序。不需要手動介入。因此，建議您DOWN盡可能嘗試將節點設定為 。
- 停用佇列（分割區）或停止特定分割區中的所有靜態節點：INACTIVE使用 scontrol update partition=*queue name* state=inactive命令將特定佇列設為 。這樣做會終止分割區中的所有執行個體備份節點。
- 啟用佇列（分割區）：INACTIVE使用 scontrol update partition=*queue name* state=up命令將特定佇列設為 。

擴展行為和調整

以下是正常擴展工作流程的範例：

- 排程器會收到需要兩個節點的任務。
- 排程器會將兩個節點轉換為 POWER_UP 狀態，並使用節點名稱ResumeProgram呼叫（例如 queue1-dy-c5xlarge-[1-2]）。
- ResumeProgram 會啟動兩個EC2執行個體，並指派 的私有 IP 地址和主機名稱queue1-dy-c5xlarge-[1-2]，等待 ResumeTimeout（預設期間為 60 分鐘（1 小時）），然後再重設節點。
- 已設定執行個體並加入叢集。任務開始在執行個體上執行。

- 任務已完成。
- 設定 經過 SuspendTime（設定為 scaledown_idletime）後，排程器會將執行個體置於 POWER_SAVING 狀態。排程器queue1-dy-c5xlarge-[1-2]進入 POWER_DOWN 狀態SuspendProgram，並使用節點名稱呼叫。
- SuspendProgram 會呼叫兩個節點。節點會保持 POWER_DOWN 狀態，例如，在 idle% 中保持 SuspendTimeout（預設期間為 120 秒（2 分鐘））。在 clustermgtd偵測到節點正在關閉電源後，它會終止備份執行個體。然後，它會queue1-dy-c5xlarge-[1-2]將 設定為閒置狀態，並重設私有 IP 地址和主機名稱，以便再次為未來的任務開啟電源。

現在，如果發生錯誤，且特定節點的執行個體因某種原因而無法啟動，則會發生以下情況。

- 排程器收到需要兩個節點的任務。
- 排程器會將兩個雲端爆量節點置於 POWER_UP 狀態ResumeProgram，並使用節點名稱呼叫（例如 queue1-dy-c5xlarge-[1-2]）。
- ResumeProgram 只會啟動一（1）個EC2執行個體並設定 queue1-dy-c5xlarge-1，但無法啟動 的執行個體queue1-dy-c5xlarge-2。
- queue1-dy-c5xlarge-1 不會受到影響，且會在達到 POWER_UP 狀態後上線。
- queue1-dy-c5xlarge-2 會處於 POWER_DOWN 狀態，且任務會自動重新排入佇列，因為 Slurm 偵測到節點失敗。
- queue1-dy-c5xlarge-2 在 SuspendTimeout（預設值為 120 秒（2 分鐘））後，即可供使用。同時，任務會重新排入佇列，並可在另一個節點上開始執行。
- 上述程序會重複進行，直到任務可以在可用的節點上執行，而不會發生失敗。

如有需要，有兩個時間參數可以調整。

- ResumeTimeout（預設值為 60 分鐘（1 小時））：ResumeTimeout 控制時間 Slurm 會等待，再將節點置於停機狀態。
  - 如果您的安裝前/後程序耗時近此長，延長此時間可能很有用。
  - 如果發生問題，這也是在取代或重設節點之前 AWS ParallelCluster 等待的時間上限。如果在啟動或設定期間發生任何錯誤，運算節點會自行終止。接下來，AWS ParallelCluster 程序也會在看到執行個體終止時取代節點。
- SuspendTimeout（預設值為 120 秒（2 分鐘））：SuspendTimeout控制節點放回系統並準備好再次使用的速度。
  - 較短SuspendTimeout表示節點會更快重設，且 Slurm 能夠嘗試更頻繁地啟動執行個體。

- 較長時間SuspendTimeout會使失敗的節點重設速度更慢。與此同時，Slurm 輪胎以使用其他節點。如果 SuspendTimeout 超過幾分鐘，Slurm 會嘗試循環瀏覽系統中的所有節點。較長SuspendTimeout的時間可能有利於大規模系統（超過 1，000 個節點），以降低 Slurm 透過經常重新佇列失敗的任務。
- 請注意， SuspendTimeout不代表終止節點備份執行個體的 AWS ParallelCluster 等待時間。power down 節點的備份執行個體會立即終止。終止程序通常會完成幾分鐘。不過，在此期間，節點會保持關機狀態，且無法在排程器中使用。

新架構的日誌

下列燈光包含多個佇列架構的金鑰日誌。搭配 Amazon CloudWatch Logs 使用的日誌串流名稱具有 格式 *{hostname}.{instance_id}.{logIdentifier}*，其中 *logIdentifier* 遵循日誌名稱。如需詳細資訊，請參閱與 Amazon CloudWatch Logs 整合。

- ResumeProgram:

  /var/log/parallelcluster/slurm_resume.log (slurm_resume)
- SuspendProgram:

  /var/log/parallelcluster/slurm_suspend.log (slurm_suspend)
- clustermgtd:

  /var/log/parallelcluster/clustermgtd.log (clustermgtd)
- computemgtd:

  /var/log/parallelcluster/computemgtd.log (computemgtd)
- slurmctld:

  /var/log/slurmctld.log (slurmctld)
- slurmd:

  /var/log/slurmd.log (slurmd)

常見問題以及如何除錯：

無法啟動、開啟電源或加入叢集的節點：

- 動態節點：

  - 檢查ResumeProgram日誌，查看是否ResumeProgram曾經使用節點呼叫。如果沒有，請檢查slurmctld日誌以判斷 Slurm 曾經嘗試ResumeProgram使用節點呼叫。請注意， 上的許可不正確ResumeProgram可能會導致其無訊息失敗。
  - 如果 ResumeProgram 已呼叫 ，請檢查是否已為節點啟動執行個體。如果無法啟動執行個體，則應有清楚的錯誤訊息，說明執行個體為何無法啟動。
  - 如果執行個體已啟動，在引導程序期間可能會出現一些問題。從ResumeProgram日誌中尋找對應的私有 IP 地址和執行個體 ID，並在日誌中查看特定執行個體的對應引導 CloudWatch 日誌。
- 靜態節點：
  - 檢查clustermgtd日誌，查看是否針對節點啟動執行個體。如果沒有，執行個體無法啟動的原因應該會明確出現錯誤。
  - 如果執行個體已啟動，則在引導程序期間會有一些問題。從clustermgtd日誌中尋找對應的私有 IP 和執行個體 ID，並在日誌中查看特定執行個體的對應引導 CloudWatch 日誌。

節點意外更換或終止，節點失敗

- 節點意外更換/終止
  - 在大多數情況下， 會clustermgtd處理所有節點維護動作。若要檢查是否已clustermgtd取代或終止節點，請檢查clustermgtd日誌。
  - 如果clustermgtd已取代或終止節點，則應該會出現一則訊息，指出動作的原因。如果原因與排程器相關（例如，節點為 DOWN），請查看slurmctld日誌以取得更多詳細資訊。如果原因EC2相關，請使用 工具來檢查該執行個體的狀態或日誌。例如，您可以檢查執行個體是否有排程事件或EC2運作狀態檢查失敗。
  - 如果 clustermgtd未終止節點，請檢查是否已computemgtd終止節點，或是否已EC2終止執行個體以擷取 Spot 執行個體。
- 節點失敗
  - 在大多數情況下，如果節點失敗，任務會自動重新排入佇列。在slurmctld日誌中查看任務或節點失敗的原因，並從中分析情況。

更換或終止執行個體時失敗，關閉節點時失敗

- 一般而言， clustermgtd會處理所有預期的執行個體終止動作。請查看clustermgtd日誌，了解為何無法取代或終止節點。
- 對於失敗的動態節點scaledown_idletime，請在SuspendProgram日誌中查看 的程式是否slurmctld以特定節點作為引數。請注意，實際上SuspendProgram不會執行任何特定動作。相

反地，它只會在呼叫時記錄。所有執行個體終止和NodeAddr重設都由 完成clustermgtd。Slurm會在 IDLE之後將節點放入 SuspendTimeout。

其他問題

- AWS ParallelCluster 不會進行任務配置或擴展決策。它會嘗試根據 啟動、終止和維護資源 Slurm的指示。

  如需任務配置、節點配置和擴展決策的問題，請參閱slurmctld日誌是否有錯誤。

Torque Resource Manager (**torque**)

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。您可以在 和 2.11.4 之前的版本中繼續使用這些版本，但它們不符合 AWS 服務與 AWS 支援團隊未來更新或故障診斷支援的資格。

AWS ParallelCluster 2.11.4 版及更早版本使用 Torque Resource Manager 6.1.2. 如需關於 Torque Resource Manager 6.1.2，請參閱 http://docs.adaptivecomputing.com/torque/6-1-2/releaseNotes/torquerelnote.htm。如需文件，請參閱 http://docs.adaptivecomputing.com/torque/6-1-2/adminGuide/torque.htm。如需來源碼，請參閱「https://github.com/adaptivecomputing/torque/tree/6.1.2」。

AWS ParallelCluster 2.4.0 版及更早版本使用 Torque Resource Manager 6.0.2. 如需版本備註，請參閱 http://docs.adaptivecomputing.com/torque/6-0-2/releaseNotes/torqueReleaseNotes6.0.2.pdf。如需文件，請參閱 http://docs.adaptivecomputing.com/torque/6-0-2/adminGuide/help.htm。如需來源碼，請參閱「https://github.com/adaptivecomputing/torque/tree/6.0.2」。

AWS Batch (**awsbatch**)

如需 的相關資訊 AWS Batch，請參閱 AWS Batch。如需文件，請參閱 AWS Batch 使用者指南 。

AWS ParallelCluster CLI 的 命令 AWS Batch

當您使用awsbatch排程器時， AWS ParallelCluster CLI 的命令 AWS Batch 會自動安裝在 AWS ParallelCluster 主機節點中。CLI 使用 AWS Batch API 操作並允許下列操作：

- 提交和管理任務。

- 監控任務、佇列和主機。
- 鏡像傳統排程器命令。

> **Important**
>
> AWS ParallelCluster 不支援 GPU的任務 AWS Batch。如需詳細資訊，請參閱GPU任務 。

主題

- awsbsub
- awsbstat
- awsbout
- awsbkill
- awsbqueues
- awsbhosts

awsbsub

將任務提交至叢集的任務佇列。

```
awsbsub [-h] [-jn JOB_NAME] [-c CLUSTER] [-cf] [-w WORKING_DIR]
        [-pw PARENT_WORKING_DIR] [-if INPUT_FILE] [-p VCPUS] [-m MEMORY]
        [-e ENV] [-eb ENV_DENYLIST] [-r RETRY_ATTEMPTS] [-t TIMEOUT]
        [-n NODES] [-a ARRAY_SIZE] [-d DEPENDS_ON]
        [command] [arguments [arguments ...]]
```

> **Important**
>
> AWS ParallelCluster 不支援 GPU的任務 AWS Batch。如需詳細資訊，請參閱GPU任務 。

定位引數

command

提交任務（指定的命令必須可用於運算執行個體）或要傳輸的檔案名稱。另請參閱--command-file。

arguments

(選用) 指定命令或命令檔案的引數。

具名引數

-jn *JOB_NAME*, --job-name *JOB_NAME*

為任務命名。第一個字元必須是字母或數字。任務名稱可以包含字母（大小寫）、數字、連字號和底線，長度上限為 128 個字元。

-c *CLUSTER*, --cluster *CLUSTER*

指定要使用的叢集。

-cf, --command-file

指出命令是要傳輸至運算執行個體的檔案。

預設：False

-w *WORKING_DIR*, --working-dir *WORKING_DIR*

指定要做為任務工作目錄的資料夾。如果未指定工作目錄，任務會在使用者主目錄的job-*<AWS_BATCH_JOB_ID>*子資料夾中執行。您可以使用此參數或 --parent-working-dir 參數。

-pw *PARENT_WORKING_DIR*, --parent-working-dir *PARENT_WORKING_DIR*

指定任務工作目錄的父資料夾。如果未指定父工作目錄，則會預設為使用者的主目錄。系統會在上層工作目錄中建立一個名為 job-*<AWS_BATCH_JOB_ID>* 的子資料夾。您可以使用此參數或 --working-dir 參數。

-if *INPUT_FILE*, --input-file *INPUT_FILE*

在任務的工作目錄中指定要傳輸至運算執行個體的檔案。您可以指定多個輸入檔案參數。

-p *VCPUS*, --vcpus *VCPUS*

指定 vCPUs 要保留給容器的 數目。與 搭配使用時–nodes，它會識別每個節點 vCPUs 的 數目。

預設：1

-m *MEMORY*, --memory *MEMORY*

指定要提供給任務的記憶體的硬性限制 (以 MiB 為單位)。如果您的任務嘗試超過此處指定的記憶體限制，則任務會結束。

預設：128

-e *ENV*, --env *ENV*

指定以逗號分隔的清單，其中列出要匯出至任務環境的環境變數名稱。若要匯出所有環境變數，請指定「所有」。請注意，「所有」環境變數清單不包含-env-blacklist參數中列出的變數，或以PCLUSTER_*或 AWS_* 字首開頭的變數。

-eb *ENV_DENYLIST*, --env-blacklist *ENV_DENYLIST*

指定以逗號分隔的清單，其中列出不匯出至任務環境的環境變數名稱。根據預設，不會匯出HOME、PWD、USER、PATH、LD_LIBRARY_PATH、TERM 和 TERMCAP。

-r *RETRY_ATTEMPTS*, --retry-attempts *RETRY_ATTEMPTS*

指定將任務移至 RUNNABLE 狀態的次數。您可以指定嘗試 1 至 10 次。如果嘗試的值大於 1，則會在任務失敗時重試任務，直到其移至指定次數RUNNABLE的狀態為止。

預設：1

-t *TIMEOUT*, --timeout *TIMEOUT*

以秒為單位指定持續時間（從任務嘗試的startedAt時間戳記中測量），此後會在任務尚未完成時 AWS Batch 終止任務。逾時值必須至少為 60 秒。

-n *NODES*, --nodes *NODES*

指定要為任務保留的節點數目。指定此參數的值，以啟用多節點平行提交。

> **Note**
>
> 參數cluster_type設定為 時，不支援多節點平行作業spot。

-a *ARRAY_SIZE*, --array-size *ARRAY_SIZE*

指出陣列的大小。您可指定介於 2 到 10,000 之間的值。如果您對任務指定陣列屬性，它會變成陣列任務。

-d *DEPENDS_ON*, --depends-on *DEPENDS_ON*

指定以分號分隔的清單，其中列出任務的相依性。一個任務可以取決於最多 20 個任務。您可以指定SEQUENTIAL類型相依性，而無需指定陣列任務的任務 ID。序列相依性允許每個子陣列任務循序完成，從索引 0 開始。您也可以指定 N_TO_N 類型相依性，以及陣列任務的任務 ID。N_TO_N 相

依性表示，此任務的每個索引子系必須等待各相依性對應的索引子系完成後，才能開始。此參數的語法為 "jobId=*<string>*，type=*<string>*;...".

awsbstat

顯示在叢集的任務佇列中提交的任務。

```
awsbstat [-h] [-c CLUSTER] [-s STATUS] [-e] [-d] [job_ids [job_ids ...]]
```

定位引數

job_ids

指定IDs要在輸出中顯示的以空格分隔的工作清單。如果該任務為任務陣列，則會顯示所有子任務。如果請求單一任務，則會以詳細版本顯示它。

具名引數

-c *CLUSTER*, --cluster *CLUSTER*

指出要使用的叢集。

-s *STATUS*, --status *STATUS*

指定以逗號分隔的清單，其中列出要包含的任務狀態。預設任務狀態為「作用中」。可接受的值為：SUBMITTED、PENDING、RUNNABLE、STARTING、RUNNING、SUCCEEDED、FAILED 和 ALL。

預設：“SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING”

-e, --expand-children

展開具有子項 (陣列和多節點平行) 的任務。

預設：False

-d, --details

顯示任務詳細資訊。

預設：False

awsbout

顯示特定任務的輸出。

```
awsbout [ -h ] [ -c CLUSTER ] [ -hd HEAD ] [ -t TAIL ] [ -s ] [ -sp STREAM_PERIOD ] job_id
```

定位引數

job_id

指定任務 ID。

具名引數

-c *CLUSTER*, --cluster *CLUSTER*

指出要使用的叢集。

-hd *HEAD*, --head *HEAD*

取得第一個 *HEAD* 任務輸出的 行。

-t *TAIL*, --tail *TAIL*

取得任務輸出的最後一個 <tail> 行。

-s, --stream

取得任務輸出，然後等待產生額外的輸出。此引數可與 –tail 搭配使用，從任務輸出的最新 <tail> 行開始。

預設：False

-sp *STREAM_PERIOD*, --stream-period *STREAM_PERIOD*

設定串流期間。

預設：5

awsbkill

取消或終止叢集中提交的任務。

```
awsbkill [ - h ] [ - c CLUSTER ] [ - r REASON ] job_ids [ job_ids ... ]
```

定位引數

job_ids

指定IDs要取消或終止之以空格分隔的工作清單。

具名引數

-c *CLUSTER*, --cluster *CLUSTER*

指出要使用的叢集名稱。

-r *REASON*, --reason *REASON*

指出要附加至任務的訊息，說明任務取消的原因。

預設：“Terminated by the user”

awsbqueues

顯示與叢集相關聯的任務佇列。

```
awsbqueues [ - h ] [ - c CLUSTER ] [ - d ] [ job_queues [ job_queues ... ]]
```

定位引數

job_queues

指定以空格分隔的清單，其中列出要顯示的佇列名稱。如果請求單一佇列，則會以詳細版本顯示它。

具名引數

-c *CLUSTER*, --cluster *CLUSTER*

指定要使用的叢集名稱。

-d, --details

指出是否顯示佇列的詳細資訊。

預設：False

awsbhosts

顯示屬於叢集運算環境的主機。

```
awsbhosts [ - h ] [ - c CLUSTER ] [ - d ] [ instance_ids [ instance_ids ... ]]
```

定位引數

instance_ids

指定執行個體 的空間分隔清單IDs。如果請求單一執行個體，則會以詳細版本顯示它。

具名引數

-c *CLUSTER*, --cluster *CLUSTER*

指定要使用的叢集名稱。

-d, --details

指出是否顯示主機的詳細資訊。

預設：False

AWS ParallelCluster 資源和標記

透過 AWS ParallelCluster ，您可以建立標籤來追蹤和管理 AWS ParallelCluster 資源。您可以在叢集組態檔案的 tags區段中定義 AWS CloudFormation 要建立並傳播至所有叢集資源的標籤。您也可以使用 AWS ParallelCluster 自動產生的標籤來追蹤和管理資源。

當您建立叢集時，叢集及其資源會標記本節中定義的 AWS ParallelCluster 和 AWS 系統標籤。

AWS ParallelCluster 會將標籤套用至叢集執行個體、磁碟區和資源。若要識別叢集堆疊， 會將 AWS 系統標籤 AWS CloudFormation 套用至叢集執行個體。若要識別叢集EC2啟動範本，請將系統標籤EC2套用至執行個體。您可以使用這些標籤來檢視和管理 AWS ParallelCluster 資源。

您無法修改 AWS 系統標籤。為了避免影響 AWS ParallelCluster 功能，請勿修改 AWS ParallelCluster 標籤。

以下是 AWS ParallelCluster 資源 AWS 的系統標籤範例。您無法修改它們。

```
"aws:cloudformation:stack-name"="parallelcluster-clustername-
MasterServerSubstack-ABCD1234EFGH"
```

以下是套用至資源的 AWS ParallelCluster 標籤範例。請勿修改它們。

```
"aws-parallelcluster-node-type"="Master"
```

```
"Name"="Master"
```

```
"Version"="2.11.9"
```

您可以在 的 EC2區段中檢視這些標籤 AWS Management Console。

檢視標籤

1. 在 導覽EC2主控台https://console.aws.amazon.com/ec2/。
2. 若要檢視所有叢集標籤，請在導覽窗格中選擇標籤。
3. 若要依執行個體檢視叢集標籤，請在導覽窗格中選擇執行個體。
4. 選取叢集執行個體。
5. 選擇執行個體詳細資訊中的管理標籤索引標籤，並檢視標籤。
6. 在執行個體詳細資訊中選擇 Storage 索引標籤。
7. 選取磁碟區 ID 。
8. 在磁碟區 中，選擇磁碟區。
9. 選擇磁碟區詳細資訊中的標籤索引標籤，並檢視標籤。

AWS ParallelCluster 主節點執行個體標籤

金鑰	標籤值
ClusterName	*clustername*
Name	Master
Application	parallelcluster- *clustername*

金鑰	標籤值
aws:ec2launchtemplate:id	*lt-1234567890abcdef0*
aws:ec2launchtemplate:version	*1*
aws-parallelcluster-node-type	Master
aws:cloudformation:stack-name	parallelcluster- *clustername* -MasterServerSubstack- *ABCD1234EFGH*
aws:cloudformation:logical-id	MasterServer
aws:cloudformation:stack-id	arn:aws:cloudformation: *region-id* :*ACCOUNTID* :stack/parallelcluster- *clustername* -MasterServerSubstack- *ABCD1234EFGH* /*1234abcd-12ab-12ab-12ab-1234567890abcdef0*
Version	*2.11.9*

AWS ParallelCluster 頭節點根磁碟區標籤

標籤鍵	標籤值
ClusterName	*clustername*
Application	parallelcluster- *clustername*
aws-parallelcluster-node-type	Master

AWS ParallelCluster 運算節點執行個體標籤

金鑰	標籤值
ClusterName	*clustername*

金鑰	標籤值
aws-parallelcluster-node-type	Compute
aws:ec2launchtemplate:id	*lt-1234567890abcdef0*
aws:ec2launchtemplate:version	*1*
QueueName	*queue-name*
Version	*2.11.9*

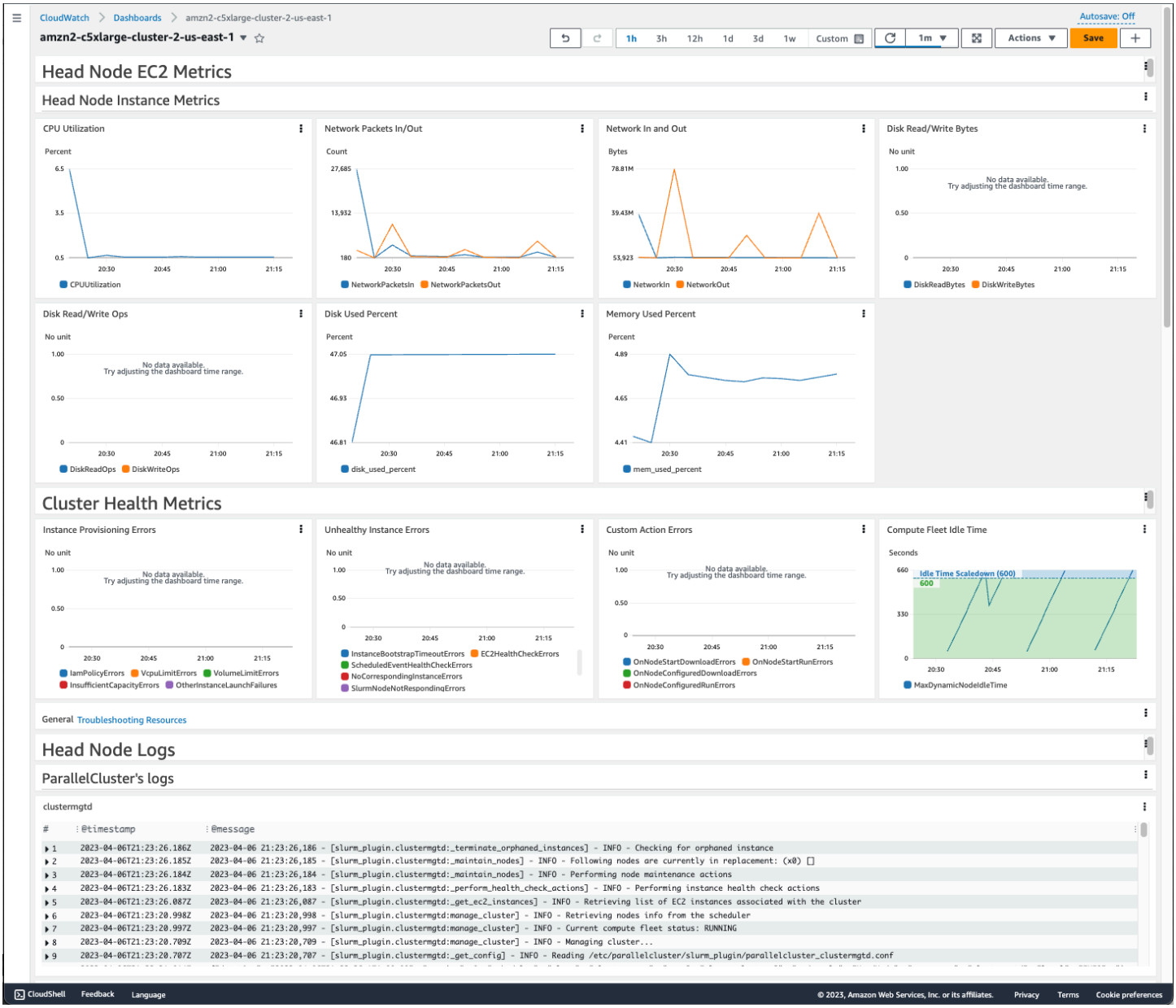
AWS ParallelCluster 運算節點根磁碟區標籤

標籤鍵	標籤值
ClusterName	*clustername*
Application	parallelcluster- *clustername*
aws-parallelcluster-node-type	Compute
QueueName	*queue-name*
Version	*2.11.9*

Amazon CloudWatch 儀表板

從 2.10.0 AWS ParallelCluster 版開始，在建立叢集時建立 Amazon CloudWatch 儀表板。這可讓您更輕鬆地監控叢集中的節點，以及檢視儲存在 Amazon CloudWatch Logs 中的日誌。儀表板的名稱為 parallelcluster-*ClusterName*-*Region*。*ClusterName* 是叢集的名稱，而 *Region* 是叢集 AWS 區域 的 。您可以在主控台中存取儀表板，也可以開啟 https://console.aws.amazon.com/cloudwatch/home?region=*Region*#dashboards:name=parallelcluster-*ClusterName*。

下圖顯示叢集的範例 CloudWatch 儀表板。



儀表板的第一個區段會顯示 Head Node EC2指標的圖形。如果您的叢集具有共用儲存體，則下一節會顯示共用儲存體指標。最後一個區段會列出依 ParallelCluster日誌、排程器日誌、NICEDCV整合日誌和系統日誌分組的 Head Node Logs。

如需 Amazon CloudWatch 儀表板的詳細資訊，請參閱 Amazon 使用者指南 中的使用 Amazon CloudWatch 儀表板。 CloudWatch

如果您不想建立 Amazon CloudWatch 儀表板，則必須完成下列步驟：首先，將[dashboard]區段新增至您的組態檔案，然後將該區段的名稱新增為[cluster]區段 中dashboard_settings設定的值。在[dashboard]區段 中，設定 enable = false。

例如，如果您的 [dashboard] 區段已命名，myDashboard而您的 [cluster]區段已命名為 myCluster，則您的變更會像這樣。

```
[cluster MyCluster]
dashboard_settings = MyDashboard
...

[dashboard MyDashboard]
enable = false
```

與 Amazon CloudWatch Logs 整合

從 2.6.0 AWS ParallelCluster 版開始，常用日誌預設會儲存在 CloudWatch 日誌中。如需 CloudWatch 日誌的詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南 。若要設定 CloudWatch 日誌整合，請參閱 [cw_log]一節和 cw_log_settings設定。

會為每個具有名稱的叢集建立日誌群組 /aws/parallelcluster/*cluster-name*（例如 /aws/parallelcluster/testCluster）。每個節點上的每個日誌（如果路徑包含 ，則為一組日誌*）都有名為 的日誌串流*{hostname}*.*{instance_id}*.*{logIdentifier}*。（例如 ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher。）日誌資料 CloudWatch 由CloudWatch 代理程式 傳送至 ，該代理程式會在所有叢集執行個體root上執行。

從 2.10.0 AWS ParallelCluster 版開始，在建立叢集時建立 Amazon CloudWatch 儀表板。此儀表板可讓您輕鬆檢閱儲存在日誌中的 CloudWatch 日誌。如需詳細資訊，請參閱Amazon CloudWatch 儀表板。

此清單包含 *logIdentifier* 和可用於平台、排程器和節點的日誌串流路徑。

可用於平台、排程器和節點的日誌串流

平台	排程器	節點	日誌串流
amazon centos ubuntu	awsbatc slurm	HeadNc	dcv-authenticator： /var/log/parallelcluster/pcluster_dcv_authenticator.log dcv-ext-authenticator: /var/log/parallelcluster/pcluster_dcv_connect.log dcv-agent： /var/log/dcv/agent.*.log

平台	排程器	節點	日誌串流
			dcv-xsession： /var/log/dcv/dcv-xsession.*.log dcv-server： /var/log/dcv/server.log dcv-session-launcher: /var/log/dcv/sessionlauncher.log Xdcv： /var/log/dcv/Xdcv.*.log cfn-init： /var/log/cfn-init.log 廚師-用戶端： /var/log/chef-client.log
amazon centos ubuntu	awsbatc slurm	Comput eet HeadNc	cloud-init： /var/log/cloud-init.log 受監控： /var/log/supervisord.log
amazon centos ubuntu	slurm	Comput eet	cloud-init-output: /var/log/cloud-init-output.log computingmgtd： /var/log/parallelcluster/co mputemgtd slurmd： /var/log/slurmd.log
amazon centos ubuntu	slurm	HeadNc	clustermgtd： /var/log/parallelcluster/clustermgtd slurm_resume： /var/log/parallelcluster/slurm_resum e.log slurm_suspend： /var/log/parallelcluster/slurm_suspe nd.log slurmctld： /var/log/slurmctld.log
amazon centos	awsbatc slurm	Comput eet HeadNc	系統訊息： /var/log/messages

平台	排程器	節點	日誌串流
ubuntu	awsbatc slurm	Comput eet HeadNc	syslog：/var/log/syslog

使用 的叢集中的任務會 AWS Batch 儲存 CloudWatch 日誌中達到 RUNNING、 SUCCEEDED或 FAILED 狀態的任務輸出。日誌群組為 /aws/batch/job，日誌串流名稱格式為 *jobDefinitionName*/default/*ecs_task_id*。根據預設，這些日誌設定為永遠不會過期，但您可以修改保留期間。如需詳細資訊，請參閱 Amazon Logs 使用者指南 中的變更 CloudWatch 日誌中的日誌資料保留。CloudWatch

> **Note**
>
> chef-client2.9.0 cloud-init-output clustermgtd AWS ParallelCluster 版中slurm_suspend已新增 slurm_resume、、computemgtd、、 和 。對於 2.6.0 AWS ParallelCluster 版，/var/log/cfn-init-cmd.log（cfn-init-cmd）和 /var/log/cfn-wire.log（cfn-wire）也儲存在 CloudWatch 日誌中。

Elastic Fabric Adapter

Elastic Fabric Adapter（EFA）是具有 OS-bypass 功能的網路裝置，可低延遲地與相同子網路上的其他執行個體進行網路通訊。EFA 使用 Libfabric 公開，且應用程式可以使用 Messaging Passing Interface（MPI）來使用。

若要EFA搭配 使用 AWS ParallelCluster，請將 行新增至 enable_efa = true[queue]區段。

若要檢視支援 的EC2執行個體清單EFA，請參閱 Amazon EC2 適用於 Linux 執行個體的使用者指南 中的支援執行個體類型。

如需enable_efa設定的詳細資訊，請參閱 一節enable_efa中的 。 [queue]

應使用叢集置放群組以充分減少執行個體之間的延遲。如需詳細資訊，請參閱 placement 和 placement_group。

如需詳細資訊，請參閱 Amazon EC2使用者指南中的 Elastic Fabric Adapter，以及使用彈性結構轉接器擴展HPC工作負載，以及 AWS ParallelClusterAWS 開放原始碼部落格中的 。

> **Note**
>
> 根據預設，Ubuntu 分佈啟用 ptrace（程序追蹤）保護。從 AWS ParallelCluster 2，6.0 開始 ptrace 會停用保護，讓 Libfabric 正常運作。如需詳細資訊，請參閱 Amazon EC2使用者指南中的停用 ptrace 保護。

> **Note**
>
> 在 2.10.1 AWS ParallelCluster 版中新增了對 Arm 型 Graviton2 執行個體EFA的支援。

Intel Select 解決方案

AWS ParallelCluster 可作為 Intel Select 模擬和建模解決方案。設定經過驗證，以符合 Intel HPC Platform Specification 設定的標準、使用特定 Intel 執行個體類型，並設定為使用 Elastic Fabric Adapter（EFA）網路介面。 AWS ParallelCluster 是第一個符合 Intel Select Solutions 程式需求的雲端解決方案。支援的執行個體類型包括 `c5n.18xlarge`、 `m5n.24xlarge`和 `r5n.24xlarge`。與 Intel Select Solutions 標準相容的範例組態提供如下。

Example Intel Select Solutions 組態

```
[global]
update_check = true
sanity_check = true
cluster_template = intel-select-solutions

[aws]
aws_region_name = <Your AWS ##>

[scaling demo]
scaledown_idletime = 5

[cluster intel-select-solutions]
key_name = <Your SSH key name>
base_os = centos7
scheduler = slurm
```

```
enable_intel_hpc_platform = true
master_instance_type = c5.xlarge
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = c5n,m5n,r5n
master_root_volume_size = 200
compute_root_volume_size = 80

[queue c5n]
compute_resource_settings = c5n_i1
enable_efa = true
placement_group = DYNAMIC

[compute_resource c5n_i1]
instance_type = c5n.18xlarge
max_count = 5

[queue m5n]
compute_resource_settings = m5n_i1
enable_efa = true
placement_group = DYNAMIC

[compute_resource m5n_i1]
instance_type = m5n.24xlarge
max_count = 5

[queue r5n]
compute_resource_settings = r5n_i1
enable_efa = true
placement_group = DYNAMIC

[compute_resource r5n_i1]
instance_type = r5n.24xlarge
max_count = 5
```

如需 AWS ParallelCluster 和 Intel HPC平台規格的詳細資訊，請參閱 Intel HPC平台規格。

啟用 Intel MPI

Intel MPI 可在 上使用 AWS ParallelCluster AMIs。若要使用 Intel MPI，您必須確認並接受 Intel 簡化軟體授權 的條款。根據預設，Open MPI 會放置在 路徑上。若要啟用 Intel MPI而非 Open MPI，您必

須先載入 Intel MPI模組。然後，您需要使用 安裝最新版本`module load intelmpi`。模組的確切名稱會隨著每次更新而改變。要查看哪些模組可用，請執行 `module avail`。輸出如下。

```
$ module avail

--------------------------------------- /usr/share/Modules/modulefiles
 ----------------------------------------
dot                        libfabric-aws/1.8.1amzn1.3 module-info                null
                     use.own
module-git                 modules                    openmpi/4.0.2

-------------------------------------------------- /etc/modulefiles
 --------------------------------------------------

--------------------------------- /opt/intel/impi/2019.7.217/intel64/modulefiles
 ---------------------------------
intelmpi
```

```
$ module load intelmpi
```

要查看哪些模組已載入，請執行 `module list`。

```
$ module list
Currently Loaded Modulefiles:
  1) intelmpi
```

若要驗證 Intel MPI 是否已啟用，請執行 `mpirun --version`。

```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2019 Update 7 Build 20200312 (id:
 5dc2dd3e9)
Copyright 2003-2020, Intel Corporation.
```

載入 Intel MPI模組後，多個路徑會變更為使用 Intel MPI工具。若要執行 Intel MPI工具編譯的程式碼，請先載入 Intel MPI模組。

> **Note**
>
> Intel MPI 與 AWS Graviton 型執行個體不相容。

> **Note**
>
> 在 2.5.0 AWS ParallelCluster 版之前，Intel MPI 在中國（北京）和中國（寧夏）區域無法使用 AWS ParallelCluster AMIs。

Intel HPC平台規格

AWS ParallelCluster 符合 Intel HPC Platform Specification。Intel HPC Platform Specification 提供一組運算、網狀架構、記憶體、儲存體和軟體需求，協助實現與HPC工作負載的高品質標準和相容性。如需詳細資訊，請參閱 Intel HPC Platform Specification 和與 Intel HPC Platform Specification 相容的應用程式驗證。

若要符合 Intel HPC平台規格，必須符合下列要求：

- 作業系統必須 CentOS 7（`base_os = centos7`）。
- 運算節點的執行個體類型必須具有 Intel CPU和至少 64 GB 的記憶體。對於執行個體類型c5系列，這表示執行個體類型必須至少為 `c5.9xlarge`（`compute_instance_type = c5.9xlarge`）。
- 主節點必須至少有 200 GB 的儲存空間。
- 必須接受 Intel Parallel Studio 的最終使用者授權合約 (`enable_intel_hpc_platform = true`)。
- 每個運算節點至少必須有 80 GB 的儲存空間 (`compute_root_volume_size = 80`)。

儲存體可以是本機或網路上（NFS從主機節點、Amazon EBS或 FSx Lustre 共用），而且可以共用。

Arm Performance 程式庫

從 2.10.1 AWS ParallelCluster 版開始，Arm Performance 程式庫可在 AWS ParallelCluster AMIs 上用於 `alinux2`、`ubuntu1804`、 `centos8`和 `base_os` 設定`ubuntu2004`的值。Arm Performance 程式庫為 Arm 處理器上的高效能運算應用程式提供最佳化的標準核心數學程式庫。若要使用 Arm Performance 程式庫，您必須確認並接受 Arm Performance 程式庫（免費版本）- 最終使用者授權合約 的條款。如需 Arm Performance 程式庫的詳細資訊，請參閱 Free Arm Performance 程式庫 。

若要啟用 Arm Performance 程式庫，您必須先載入 Arm Performance 程式庫模組。 `Armpl-21.0.0` 需要 GCC-9.3 作為需求，當您載入`armpl/21.0.0`模組時，也會載入`gcc/9.3`模組。模組的確切名稱會隨著每次更新而改變。要查看哪些模組可用，請執行 `module avail`。然後，您需要使用 安裝最新版本`module load armpl`。輸出如下所示。

```
$ module avail

------------------------------ /usr/share/Modules/modulefiles
 --------------------------------
armpl/21.0.0         dot                  libfabric-aws/1.11.1amzn1.0
 module-git
module-info          modules              null                 openmpi/4.1.0
 use.own
```

要載入模組，請執行 `module load modulename`。您可以將此新增到用於執行 mpirun 的指令碼。

```
$ module load armpl

Use of the free of charge version of Arm Performance Libraries is subject to the terms
 and
conditions of the Arm Performance Libraries (free version) -  End User License
 Agreement
(EULA). A copy of the EULA can be found in the
'/opt/arm/armpl/21.0.0/arm-performance-libraries_21.0_gcc-9.3/license_terms' folder
```

要查看哪些模組已載入，請執行 `module list`。

```
$ module list
Currently Loaded Modulefiles:
1) /opt/arm/armpl/21.0.0/modulefiles/armpl/gcc-9.3
2) /opt/arm/armpl/21.0.0/modulefiles/armpl/21.0.0_gcc-9.3
3) armpl/21.0.0
```

若要驗證 Arm Performance 程式庫是否已啟用，請執行範例測試。

```
$ sudo chmod 777 /opt/arm/armpl/21.0.0/armpl_21.0_gcc-9.3/examples
$ cd /opt/arm/armpl/21.0.0/armpl_21.0_gcc-9.3/examples
$ make
...
Testing: no example difference files were generated.
Test passed OK
```

載入 Arm Performance Library 模組後，多個路徑會變更為使用 Arm Performance Libraries 工具。若要執行 Arm Performance Library 工具編譯的程式碼，請先載入 Arm Performance Library 模組。

> **Note**
>
> AWS ParallelCluster 2.10.1 和 2.10.4 之間的版本使用 `armpl/20.2.1`。

透過 Amazon 連線至主機節點 DCV

Amazon DCV 是一種遠端視覺化技術，可讓使用者安全地連線至託管在遠端高效能伺服器上的圖形密集型 3D 應用程式。如需詳細資訊，請參閱 Amazon DCV。

使用 `base_os = alinux2`、`base_os = ubuntu1804`或 時`base_os = centos7`，Amazon DCV 軟體會自動安裝在主機節點上`base_os = ubuntu2004`。

如果主機節點是ARM執行個體，則使用 `base_os = alinux2`、`base_os = centos7`或 時，Amazon DCV 軟體會自動安裝在其上`base_os = ubuntu1804`。

若要在主節點DCV上啟用 Amazon，`dcv_settings` 必須包含具有 `enable = master`且`base_os`必須設定為 `alinux2`、`ubuntu1804`、 `centos7`或 的`[dcv]`區段名稱`ubuntu2004`。如果主機節點是ARM執行個體，`base_os`則必須將 設定為 `alinux2`、 `centos7`或 `ubuntu1804`。如此一來， 會將叢集組態參數 AWS ParallelCluster 設定為`shared_dir`DCV伺服器儲存資料夾 。

```
[cluster custom-cluster]
...
dcv_settings = custom-dcv
...
[dcv custom-dcv]
enable = master
```

如需 Amazon DCV組態參數的詳細資訊，請參閱 `dcv_settings`。若要連線至 Amazon DCV工作階段，請使用 `pcluster dcv`命令。

> **Note**
>
> 在 DCV上的 Amazon 支援`centos8`已在 2.10.4 AWS ParallelCluster 版中移除。`centos8` 2.10.0 AWS ParallelCluster 版中已新增 DCV上的 Amazon 支援。在 2.9.0 AWS ParallelCluster 版中新增了對 Amazon DCV on AWS Graviton 型執行個體的支援。Amazon DCV on `alinux2`和 的支援`ubuntu1804`已新增至 2.6.0 AWS ParallelCluster 版。`centos7` 2.5.0 AWS ParallelCluster 版中已新增對 Amazon DCV on 的支援。

> **Note**
>
> 2.8.0 版和 2.8.1 AWS ParallelCluster 版中以 AWS Graviton 為基礎的執行個體DCV不支援 Amazon。

Amazon DCVHTTPS憑證

Amazon DCV會自動產生自我簽署憑證，以保護 Amazon DCV用戶端與 Amazon DCV 伺服器之間的流量。

若要將預設自我簽署的 Amazon DCV憑證取代為另一個憑證，請先連線至主機節點。然後，在執行 `pcluster dcv` 命令之前，將憑證和金鑰複製到 `/etc/dcv` 資料夾。

如需詳細資訊，請參閱 Amazon DCV管理員指南 中的變更TLS憑證。

授權 Amazon DCV

在 Amazon EC2執行個體上執行時，Amazon DCV 伺服器不需要授權伺服器。不過，Amazon DCV 伺服器必須定期連線至 Amazon S3 儲存貯體，以判斷是否有可用的有效授權。

AWS ParallelCluster 會自動將必要的許可新增至 `ParallelClusterInstancePolicy`。使用自訂 IAM執行個體政策時，請使用 Amazon 管理員指南 中 Amazon DCV on Amazon EC2 中所述的許可。DCV

如需疑難排解秘訣，請參閱 對 Amazon 中的問題進行故障診斷 DCV。

使用 `pcluster update`

從 2.8.0 AWS ParallelCluster 版開始， 會`pcluster update`分析用來建立目前叢集的設定，以及組態檔案中的問題設定。如果發現任何問題，則會報告這些問題，並顯示修正問題所採取的步驟。例如，如果`compute_instance_type`設定變更為不同的執行個體類型，則必須先停止運算機群，才能繼續更新。此問題會在發現時回報。如果未報告封鎖問題，系統會提示您是否要套用變更。

每個設定的文件都會定義該設定的更新政策。

更新政策：這些設定可以在更新期間變更。，更新政策：此設定可以在更新期間變更。

這些設定可以變更，而叢集可以使用 更新`pcluster update`。

更新政策：如果變更此設定，則不允許更新。

如果尚未刪除現有叢集，則無法變更這些設定。必須還原變更或刪除叢集（使用 pcluster delete），然後在舊叢集的位置建立新的叢集（使用 pcluster create）。

更新政策：更新期間不會分析此設定。

這些設定可以變更，並使用 更新叢集pcluster update。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

當運算機群存在時，無法變更這些設定。必須還原變更或停止運算機群（使用 pcluster stop）、更新（使用 pcluster update），然後建立新的運算機群（使用 pcluster start）。

更新政策：此設定無法在更新期間減少。

這些設定可以變更，但無法減少。如果必須減少這些設定，則必須刪除叢集（使用 pcluster delete），並建立新的叢集（使用 pcluster create）。

更新政策：將佇列的大小減少到目前節點數量以下，需要先停止運算機群。

這些設定可以變更，但如果變更將佇列的大小減少到目前大小以下，則必須停止運算機群（使用 pcluster stop）、更新（使用 pcluster update），然後建立新的運算機群（使用 ）pcluster start。

更新政策：減少佇列中的靜態節點數量需要先停止運算機群。

這些設定可以變更，但如果變更將佇列中的靜態節點數量減少到目前大小以下，則必須停止運算機群（使用 pcluster stop）、更新（使用 pcluster update），然後建立新的運算機群（使用 pcluster start）。

更新政策：如果變更此設定，則不允許更新。無法強制更新此設定。

如果尚未刪除現有叢集，則無法變更這些設定。必須還原變更或刪除叢集（使用 pcluster delete），然後在舊叢集的位置建立新的叢集（使用 pcluster create）。

更新政策：如果未在組態中指定受 AWS ParallelCluster 管 Amazon FSx for Lustre 檔案系統，則可以在更新期間變更此設定。

如果[cluster]fsx_settings未指定此設定，或者[fsx fs]如果 fsx_settings和 fsx-fs-id 都指定為掛載 FSx Lustre 檔案系統的現有外部 ，則可以變更此設定。

此範例示範 pcluster update，其中包含一些會封鎖更新的變更。

```
$ pcluster update
```

```
  Validating configuration file /home/username/.parallelcluster/config...
Retrieving configuration from CloudFormation for cluster test-1...
Found Changes:

#   section/parameter         old value                 new value
--  ------------------------  ------------------------  ------------------------
    [cluster default]
01* compute_instance_type     t2.micro                  c4.xlarge
02* ebs_settings              ebs2                      -

    [vpc default]
03  additional_sg             sg-0cd61884c4ad16341       sg-0cd61884c4ad11234

    [ebs ebs2]
04* shared_dir                shared                     my/very/very/long/sha...

Validating configuration update...
The requested update cannot be performed. Line numbers with an asterisk indicate
updates requiring additional actions. Please look at the details below:

#01
Compute fleet must be empty to update "compute_instance_type"
How to fix:
Make sure that there are no jobs running, then run the following command:
  pcluster stop -c $CONFIG_FILE $CLUSTER_NAME

#02
Cannot add/remove EBS Sections
How to fix:
Revert "ebs_settings" value to "ebs2"

#04
Cannot change the mount dir of an existing EBS volume
How to fix:
Revert "my/very/very/long/shared/dir" to "shared"

In case you want to override these checks and proceed with the update please
use the --force flag. Note that the cluster could end up in an unrecoverable
state.

Update aborted.
```

AMI 修補和EC2執行個體替換

為了確保所有動態啟動的叢集運算節點都以一致的方式運作， AWS ParallelCluster 請停用叢集執行個體自動作業系統更新。此外，會針對每個 AWS ParallelCluster AMIs版本 AWS ParallelCluster 及其相關聯的 建置特定的 集CLI。此特定 集合AMIs保持不變，且僅支援其為 建置的 AWS ParallelCluster 版本。 AWS ParallelCluster AMIs 發行的版本不會更新。

但是，由於出現安全問題，客戶可能會想要將修補程式新增至這些問題，AMIs然後使用修補的 更新其叢集AMI。這符合 AWS ParallelCluster 共同責任模型 。

若要檢視您目前使用之 AWS ParallelCluster CLI版本支援的特定 AWS ParallelCluster AMIs集，請執行：

```
$ pcluster version
```

然後在 AWS ParallelCluster的 GitHub 儲存庫中檢視 amis.txt。

AWS ParallelCluster 主節點是靜態執行個體，您可以手動更新。如果執行個體類型沒有執行個體存放區，則從 2.11 AWS ParallelCluster 版開始完全支援重新啟動和重新啟動主機節點。如需詳細資訊，請參閱 Amazon Linux EC2 執行個體使用者指南 中的執行個體類型與執行個體存放區磁碟區。您無法更新現有叢集AMI的 。

從 3.0.0 AWS ParallelCluster 版開始，完全支援主機節點重新啟動和重新啟動與叢集運算執行個體AMI更新。請考慮升級至最新版本以使用這些功能。

主節點執行個體更新或替換

在某些情況下，您可能需要重新啟動或重新啟動主機節點。例如，當您手動更新作業系統時，或當AWS 執行個體排程淘汰會強加主機節點執行個體重新啟動時，這是必要的。

如果您的執行個體沒有暫時性磁碟機，您可以隨時停止並再次啟動。在排程淘汰的情況下，啟動停止的執行個體會將其遷移至使用新的硬體。

同樣地，您可以手動停止和啟動沒有執行個體存放區的執行個體。對於此案例和其他沒有暫時性磁碟區的執行個體，請繼續 停止和啟動叢集的主機節點。

如果您的執行個體有暫時性磁碟機且已停止，執行個體存放區中的資料會遺失。您可以從執行個體存放區磁碟區 中找到的資料表中，判斷用於主機節點的執行個體類型是否有執行個體存放區。

下列各節說明使用執行個體與執行個體存放區磁碟區的限制。

執行個體存放區限制

搭配執行個體存放區使用 2.11 AWS ParallelCluster 版和執行個體類型的限制如下：

- 當暫時磁碟機未加密（encrypted_ephemeral 參數設定為 false或未設定）時，AWS ParallelCluster 執行個體無法在執行個體停止後開機。這是因為舊的不存在暫時性資料的資訊會寫入 ，fstab而作業系統會嘗試掛載不存在的儲存體。
- 當暫時磁碟機加密（encrypted_ephemeral 參數設定為 true）時，可以在停止後啟動 AWS ParallelCluster 執行個體，但新的暫時磁碟機未設定、掛載或可用。
- 當暫時性磁碟機加密時，AWS ParallelCluster 執行個體可以重新啟動，但無法存取舊暫時性磁碟機（在執行個體重新啟動之後），因為加密金鑰是在重新啟動時遺失的記憶體中建立的。

唯一支援的案例是執行個體重新開機，此時未加密暫時性磁碟機。這是因為磁碟機在重新開機後仍然存在，並且由於 中寫入的項目而掛回fstab。

執行個體存放區限制因應措施

首先，儲存您的資料。若要檢查是否有需要保留的資料，請檢視ephemeral_dir資料夾中的內容（/scratch 預設）。您可以將資料傳輸至根磁碟區或連接至叢集的共用儲存系統，例如 Amazon FSx、Amazon EFS或 Amazon EBS。請注意，資料傳輸至遠端儲存體可能會產生額外費用。

限制的根本原因在於 AWS ParallelCluster 邏輯中，該邏輯用於格式化和掛載執行個體存放區磁碟區。邏輯會將項目新增至 表單/etc/fstab的 ：

```
$ /dev/vg.01/lv_ephemeral ${ephemeral_dir} ext4 noatime,nodiratime 0 0
```

${ephemeral_dir} 是 pcluster 組態檔案中ephemeral_dir參數的值（預設為 /scratch）。

新增此行，以便在節點重新啟動時自動重新掛載執行個體存放區磁碟區。這很理想，因為暫時性磁碟機中的資料會透過重新啟動來持續存在。不過，暫時性磁碟機上的資料不會在啟動或停止循環期間持續存在。這表示它們已格式化並掛載，沒有資料。

唯一支援的案例是未加密暫時性磁碟機時的執行個體重新啟動。這是因為磁碟機在重新開機後仍然存留，並且因為寫入 而掛回fstab。

若要在所有其他情況下保留資料，您必須在停止執行個體之前移除邏輯磁碟區項目。例如，/dev/vg.01/lv_ephemeral在停止執行個體/etc/fstab之前從 移除 。執行此操作後，您無需掛載暫時磁碟區即可啟動執行個體。不過，執行個體存放區掛載在執行個體停止或啟動後將無法再次使用。

儲存資料然後移除fstab項目後，請繼續下一節。

停止和啟動叢集的主機節點

> **Note**
>
> 從 2.11 AWS ParallelCluster 版開始，只有在執行個體類型沒有執行個體存放區時，才支援主機節點停止和啟動。

1. 確認叢集中沒有任何執行中的任務。

   使用 時 Slurm 排程器：

   - 如果未指定 sbatch--no-requeue選項，則會重新佇列執行中的任務。
   - 如果指定 --no-requeue選項，則執行任務會失敗。

2. 請求叢集運算機群停止：

```
$ pcluster stop cluster-name
Compute fleet status is: RUNNING. Submitting status change request.
Request submitted successfully. It might take a while for the transition to
 complete.
Please run 'pcluster status' if you need to check compute fleet status
```

3. 等待運算機群狀態為 STOPPED：

```
$ pcluster status cluster-name
...
ComputeFleetStatus: STOP_REQUESTED
$ pcluster status cluster-name
...
ComputeFleetStatus: STOPPED
```

4. 對於作業系統重新啟動或執行個體重新啟動的手動更新，您可以使用 AWS Management Console 或 AWS CLI。以下是使用 的範例 AWS CLI。

```
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
  "StoppingInstances": [
    {
      "CurrentState": {
```

```
        "Name": "stopping"
        ...
      },
      "InstanceId": "i-1234567890abcdef0",
      "PreviousState": {
        "Name": "running"
        ...
      }
    }
  ]
}
$ aws ec2 start-instances --instance-ids 1234567890abcdef0
{
  "StartingInstances": [
    {
      "CurrentState": {
        "Name": "pending"
        ...
      },
      "InstanceId": "i-1234567890abcdef0",
      "PreviousState": {
        "Name": "stopped"
        ...
      }
    }
  ]
}
```

5. 啟動叢集的運算機群：

```
$ pcluster start cluster-name
Compute fleet status is: STOPPED. Submitting status change request.
Request submitted successfully. It might take a while for the transition to
 complete.
Please run 'pcluster status' if you need to check compute fleet status
```

AWS ParallelCluster CLI 命令

pcluster 和 pcluster-config AWS ParallelCluster CLI是 命令。您可以使用 pcluster 來啟動和管理 中的HPC叢集pcluster-config， AWS 雲端 以及更新您的組態。

若要使用 pcluster，您必須具有執行IAM該角色所需的許可。

```
pcluster [ -h ] ( create | update | delete | start | stop | status | list |
                  instances | ssh | dcv | createami | configure | version ) ...
 pcluster-config [-h] (convert) ...
```

主題

- pcluster
- pcluster-config

pcluster

pcluster 是主要 AWS ParallelCluster CLI命令。您可以使用 在 中pcluster啟動和管理HPC叢集 AWS 雲端。

```
pcluster [ -h ] ( create | update | delete | start | stop | status | list |
                  instances | ssh | dcv | createami | configure | version ) ...
```

引數

pcluster ***command***

可能的選擇如下：configure、create、createami、dcv、delete、instances、list、ssh、start、statu

子命令：

主題

- pcluster configure

- pcluster create
- pcluster createami
- pcluster dcv
- pcluster delete
- pcluster instances
- pcluster list
- pcluster ssh
- pcluster start
- pcluster status
- pcluster stop
- pcluster update
- pcluster version

pcluster configure

開始 AWS ParallelCluster 組態。如需詳細資訊，請參閱設定 AWS ParallelCluster。

```
pcluster configure [ -h ] [ -c CONFIG_FILE ] [ -r REGION ]
```

具名引數

-h, --help

顯示 的說明文字pcluster configure。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用之替代組態檔案的完整路徑。

預設為 ~/.parallelcluster/config。

如需詳細資訊，請參閱設定 AWS ParallelCluster。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。如果指定了此選項，則組態會略過 AWS 區域 偵測。

若要刪除 中的網路資源VPC，您可以刪除 CloudFormation 網路堆疊。堆疊名稱開頭為 "parallelclusternetworking-" 和 包含 "YYYYMMDDHHMMSS" 格式的建立時間。您可以使用 list-stacks 命令列出堆疊。

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

您可以使用 delete-stack 命令來刪除堆疊。

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

為您VPC`pcluster configure`建立的 不會在 CloudFormation 網路堆疊中建立。您可以在主控台或使用 VPC手動刪除該項目 AWS CLI。

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

pcluster create

建立新叢集。

```
pcluster create [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] [ -nr ]
                [ -u TEMPLATE_URL ] [ -t CLUSTER_TEMPLATE ]
                [ -p EXTRA_PARAMETERS ] [ -g TAGS ]
                cluster_name
```

定位引數

cluster_name

定義叢集的名稱。 AWS CloudFormation 堆疊名稱為 parallelcluster-*cluster_name*。

具名引數

-h, --help

顯示 的說明文字`pcluster create`。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。用於為新叢集選取 AWS 區域 的優先順序順序如下：

1. -r 或 --region 參數pcluster create。
2. AWS_DEFAULT_REGION 環境變數。
3. aws_region_name AWS ParallelCluster 組態檔案[aws]區段中的 設定（預設位置為 ~/.parallelcluster/config。）這是 pcluster configure命令更新的位置。
4. region AWS CLI 設定檔案（~/.aws/config.）[default]區段中的 設定

-nw, --nowait

表示在執行堆疊命令之後，不要等待堆疊事件。

預設為 False。

-nr, --norollback

發生錯誤時停用 堆疊轉返。

預設為 False。

-u *TEMPLATE_URL*, --template-url *TEMPLATE_URL*

如果建立時使用自訂 AWS CloudFormation 範本，請指定該範本URL的 。

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

指出要使用的叢集範本。

-p *EXTRA_PARAMETERS*, --extra-parameters *EXTRA_PARAMETERS*

將額外的參數新增至堆疊建立。

-g *TAGS*, --tags *TAGS*

指定要新增至堆疊的額外標籤。

當呼叫命令並開始輪詢該呼叫的狀態時，使用「Ctrl-C」結束是安全的。您可以透過呼叫 pcluster status mycluster 來返回檢視目前狀態。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster create mycluster
    Beginning cluster creation for cluster: mycluster
Info: There is a newer version 3.1.4 of AWS ParallelCluster available.
Creating stack named: parallelcluster-mycluster
Status: ComputeFleetHITSubstack - CREATE_IN_PROGRESS
$ pcluster create mycluster --tags '{ "Key1" : "Value1" , "Key2" : "Value2" }'
```

pcluster createami

（Linux/macOS）建立AMI要與 搭配使用的自訂 AWS ParallelCluster。

```
pcluster createami [ -h ] -ai BASE_AMI_ID -os BASE_AMI_OS
                   [ -i INSTANCE_TYPE ] [ -ap CUSTOM_AMI_NAME_PREFIX ]
                   [ -cc CUSTOM_AMI_COOKBOOK ] [--no-public-ip]
                   [ -post-install POST_INSTALL_SCRIPT ]
                   [ -c CONFIG_FILE ] [-t CLUSTER_TEMPLATE]
                   [--vpc-id VPC_ID] [--subnet-id SUBNET_ID]
                   [ -r REGION ]
```

必要的相依性

除了 之外 AWS ParallelCluster CLI，執行 需要下列相依性`pcluster createami`：

- Packer：從 https://developer.hashicorp.com/packer/downloads 下載最新版本。

> **Note**
>
> 在 2.8.0 AWS ParallelCluster 版之前，需要 Berkshelf（使用 安裝`gem install berkshelf`）才能使用 `pcluster createami`。

具名引數

`-h, --help`

顯示 的說明文字`pcluster createami`。

-ai *BASE_AMI_ID*, --ami-id *BASE_AMI_ID*

指定AMI用於建置 的 基礎 AWS ParallelCluster AMI。

-os *BASE_AMI_OS*, --os *BASE_AMI_OS*

指定基礎 的作業系統AMI。有效選項為：`alinux2`、`ubuntu1804`、`ubuntu2004` 和 `centos7`。

> **Note**
>
> 作業系統支援不同 AWS ParallelCluster 版本的變更：
>
> - `centos8` 2.10.4 AWS ParallelCluster 版已移除 的 支援。
> - `centos8` 已新增 的支援，且 的支援`centos6`已在 2.10.0 AWS ParallelCluster 版中移除。
> - 2.6.0 AWS ParallelCluster 版中已新增對 alinux2 的支援。
> - 2.5.0 AWS ParallelCluster 版中已新增 ubuntu1804 的支援。

-i *INSTANCE_TYPE*, --instance-type *INSTANCE_TYPE*

指定用來建立 的執行個體類型AMI。

預設為 `t2.xlarge`。

> **Note**
>
> 2.4.1 AWS ParallelCluster 版中已新增對 `--instance-type` 引數的支援。

-ap *CUSTOM_AMI_NAME_PREFIX*, --ami-name-prefix *CUSTOM_AMI_NAME_PREFIX*

指定產生的 的字首名稱 AWS ParallelCluster AMI。

預設為 `custom-ami-`。

-cc *CUSTOM_AMI_COOKBOOK*, --custom-cookbook *CUSTOM_AMI_COOKBOOK*

指定要用來建置 的食譜 AWS ParallelCluster AMI。

--post-install *POST_INSTALL_SCRIPT*

指定安裝後指令碼的路徑。路徑必須使用 `s3://`、 `https://`或 `file://`URL配置。範例如下：

- https://*bucket-name*.s3.*region*.amazonaws.com/*path*/post_install.sh
- s3://*bucket-name*/post_install.sh
- file:///*opt/project*/post_install.sh

> **Note**
>
> 2.10.0 AWS ParallelCluster 版中已新增對 `--post-install` 引數的支援。

`--no-public-ip`

請勿將公有 IP 地址與用來建立 的執行個體建立關聯AMI。根據預設，公有 IP 地址會與執行個體相關聯。

> **Note**
>
> 2.5.0 AWS ParallelCluster 版中已新增對 `--no-public-ip` 引數的支援。

`-c` *`CONFIG_FILE`*, `--config` *`CONFIG_FILE`*

指定要使用的替代組態檔案。

預設為 `~/.parallelcluster/config`。

`-t` *`CLUSTER_TEMPLATE`*, `--cluster-template` *`CLUSTER_TEMPLATE`*

指定 的 【cluster】 區段 *`CONFIG_FILE`* 用於擷取 VPC和子網路設定。

> **Note**
>
> 2.4.0 AWS ParallelCluster 版中已新增對 `--cluster-template` 引數的支援。

`--vpc-id` *`VPC_ID`*

指定VPC要用來建置 的 ID AWS ParallelCluster AMI。

> **Note**
>
> 2.5.0 AWS ParallelCluster 版中已新增對 `--vpc-id` 引數的支援。

--subnet-id *SUBNET_ID*

指定要用來建置 的子網路 ID AWS ParallelCluster AMI。

> **Note**
>
> 2.5.0 AWS ParallelCluster 版中已新增對 `--vpc-id` 引數的支援。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 `pcluster configure`命令預設為 AWS 區域 指定的 。

pcluster dcv

與在主機節點上執行的 Amazon DCV 伺服器互動。

```
pcluster dcv [ -h ] ( connect )
```

pcluster dcv *command*

可能的選擇如下：`connect`

> **Note**
>
> 作業系統支援不同 AWS ParallelCluster 版本中`pcluster dcv`命令的變更：
>
> - 2.10.0 版中`centos8` AWS ParallelCluster 已新增 上 `pcluster dcv` 命令的支援。
> - 在 2.9.0 AWS ParallelCluster 版中新增了對 AWS Graviton 型執行個體`pcluster dcv`命令的支援。
> - 2.6.0 版中`ubuntu1804` AWS ParallelCluster 已新增 上的 `pcluster dcv` 命令支援。
> - 2.5.0 版中`centos7` AWS ParallelCluster 已新增 上`pcluster dcv`命令的支援。

具名引數

-h, --help

顯示 的說明文字pcluster dcv。

子命令

pcluster dcv connect

```
pcluster dcv connect [ -h ] [ -k SSH_KEY_PATH ] [ -r REGION ] cluster_name
```

> **Important**
> 會在發出後 30 秒URL過期。如果在 URL過期之前未建立連線，請pcluster dcv connect再次執行 以產生新的 URL。

定位引數

cluster_name

指定要連接的叢集名稱。

具名引數

-h, --help

顯示 的說明文字pcluster dcv connect。

-k *SSH_KEY_PATH*, --key-path *SSH_KEY_PATH*

用於連線之金鑰的SSH金鑰路徑。

金鑰必須在 key_name 組態參數中叢集建立之時指定。此引數是選用的，但如果未指定，則金鑰預設必須可供SSH用戶端使用。例如，利用 ssh-add 將其新增至 ssh-agent。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

-s, --show-url

顯示連線至 URL Amazon DCV工作階段的一次性。指定此選項時，預設瀏覽器不會開啟。

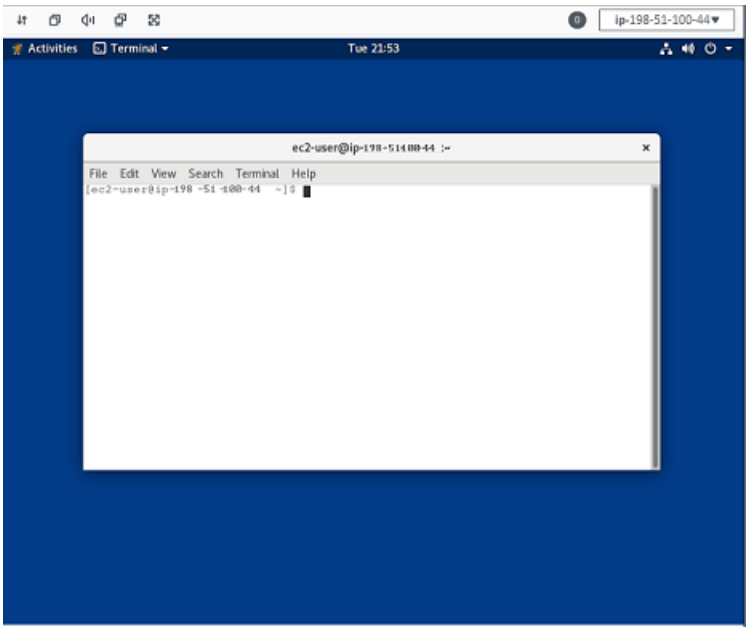
> **Note**
>
> 2.5.1 AWS ParallelCluster 版中已新增對 `--show-url` 引數的支援。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster dcv connect -k ~/.ssh/id_rsa mycluster
```

開啟預設瀏覽器以連線至在主機節點上執行的 Amazon DCV工作階段。

如果尚未啟動 Amazon DCV工作階段，則會建立新的工作階段。



pcluster delete

刪除叢集。

```
pcluster delete [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] cluster_name
```

定位引數

cluster_name

指定要刪除的叢集名稱。

具名引數

-h, --help

顯示 的說明文字pcluster delete。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

--keep-logs

刪除叢集後保留 CloudWatch 日誌資料。日誌群組會保留，直到您手動刪除為止，但日誌事件會根據retention_days設定過期。此設定預設為 14 天。

> **Note**
>
> 2.6.0 版中 AWS ParallelCluster 已新增對 **--keep-logs** 引數的支援。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

當呼叫命令並開始輪詢該呼叫的狀態時，使用「Ctrl-C」結束是安全的。您可以透過呼叫 pcluster status mycluster 來返回檢視目前狀態。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster delete -c path/to/config -r us-east-1 mycluster
Deleting: mycluster
Status: RootRole - DELETE_COMPLETE
Cluster deleted successfully.
```

若要刪除 中的網路資源VPC，您可以刪除 CloudFormation 網路堆疊。堆疊名稱開頭為 "parallelclusternetworking-" 和 包含 "YYYYMMDDHHMMSS" 格式的建立時間。您可以使用 list-stacks 命令列出堆疊。

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

您可以使用 delete-stack 命令刪除堆疊。

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

為您VPC`pcluster configure`建立的 不會在 CloudFormation 聯網堆疊中建立。您可以在主控台或使用 VPC手動刪除該項目 AWS CLI。

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

pcluster instances

將顯示叢集中所有執行個體的清單。

```
pcluster instances [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

定位引數

cluster_name

使用提供的名稱顯示叢集的執行個體。

具名引數

-h, --help

顯示 的說明文字`pcluster instances`。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster instances -c path/to/config -r us-east-1 mycluster
MasterServer         i-1234567890abcdef0
ComputeFleet         i-abcdef01234567890
```

pcluster list

顯示與 相關聯的堆疊清單 AWS ParallelCluster。

```
pcluster list [ -h ] [ -c CONFIG_FILE ] [ -r REGION ]
```

具名引數

-h, --help

顯示 的說明文字pcluster list。

--color

以顏色顯示叢集狀態。

預設為 False。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 c。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

列出名稱為 的任何 AWS CloudFormation 堆疊的名稱parallelcluster-*。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster list -c path/to/config -r us-east-1
mycluster            CREATE_IN_PROGRESS  2.11.7
myothercluster       CREATE_IN_PROGRESS  2.11.7
```

pcluster ssh

執行預先填入叢集使用者名稱和 IP 地址的 ssh 命令。任意引數會附加至 ssh 命令結尾。您可以在組態檔案的別名區段中自訂此命令。

```
pcluster ssh [ -h ] [ -d ] [ -r REGION ] cluster_name
```

定位引數

cluster_name

指定要連接的叢集名稱。

具名引數

-h, --help

顯示 的說明文字pcluster ssh。

-d, --dryrun

列印會執行和結束的命令。

預設為 False。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。預設為使用 pcluster configure 命令指定的區域。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster ssh -d mycluster -i ~/.ssh/id_rsa
SSH command: ssh ec2-user@1.1.1.1 -i /home/user/.ssh/id_rsa
```

```
$ pcluster ssh mycluster -i ~/.ssh/id_rsa
```

使用預先填入之叢集的使用者名稱和 IP 地址執行ssh命令：

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

ssh 命令的定義位於 [aliases] 區段 下的全域組態檔案中。可以自訂此命令，如下所示。

```
[ aliases ]
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

取代的變數：

CFN_USER

已選取的 base_os 使用者名稱。

MASTER_IP

主節點的 IP 地址。

ARGS

要傳遞到 ssh 命令的可選引數。

pcluster start

開始已停止的叢集運算機群。

```
pcluster start [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

定位引數

cluster_name

啟動所提供叢集名稱的運算機群。

具名引數

-h, --help

顯示 的說明文字pcluster start。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster start mycluster
Compute fleet status is: RUNNING. Submitting status change request.
Request submitted successfully. It might take a while for the transition to complete.
Please run 'pcluster status' if you need to check compute fleet status
```

此命令會將 Auto Scaling 群組參數設為下列其中一個：

- 初始組態值 (max_queue_size 和 initial_queue_size)，來自建立叢集所用的範本。
- 用來更新叢集的組態值，因為它是第一個建立的叢集。

pcluster status

提取叢集的目前狀態。

```
pcluster status [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] cluster_name
```

定位引數

cluster_name

請使用提供的名稱顯示叢集的狀態。

具名引數

-h, --help

顯示 的說明文字pcluster status。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

-nw, --nowait

表示在處理堆疊命令之後不要等待堆疊事件。

預設為 False。

使用 2.11.7 AWS ParallelCluster 版的範例：

```
$ pcluster status -c path/to/config -r us-east-1 mycluster
Status: ComputeFleetHITSubstack - CREATE_IN_PROGRESS
```

pcluster stop

停止運算機群，讓主機節點持續執行。

```
pcluster stop [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

定位引數

cluster_name

停止所提供叢集名稱的運算機群。

使用 2.11.7 AWS ParallelCluster 版的範例：

具名引數

-h, --help

顯示 的說明文字pcluster stop。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

```
$ pcluster stop mycluster
Compute fleet status is: STOPPED. Submitting status change request.
Request submitted successfully. It might take a while for the transition to complete.
Please run 'pcluster status' if you need to check compute fleet status
```

將 Auto Scaling 群組參數設定為 min/max/desired = 0/0/0，並終止運算機群。磁頭保持執行中。若要終止所有EC2資源並避免EC2收費，請考慮刪除叢集。

pcluster update

分析組態檔案，以判斷叢集是否可以安全更新。如果分析確定叢集可以更新，系統會提示您確認變更。如果分析顯示無法更新叢集，則作為衝突來源的組態設定會列舉詳細資訊。如需詳細資訊，請參閱使用 pcluster update。

```
pcluster update [ -h ] [ -c CONFIG_FILE ] [ --force ] [ -r REGION ] [ -nr ]
                [ -nw ] [ -t CLUSTER_TEMPLATE ] [ -p EXTRA_PARAMETERS ] [ -rd ]
                [ --yes ] cluster_name
```

定位引數

cluster_name

指定要更新的叢集名稱。

具名引數

-h, --help

顯示 的說明文字pcluster update。

-c *CONFIG_FILE*, --config *CONFIG_FILE*

指定要使用的替代組態檔案。

預設為 ~/.parallelcluster/config。

--force

即使一或多個設定發生封鎖變更，或需要未完成的動作（例如停止運算機群）才能繼續更新，仍啟用更新。這不應與--yes引數結合。

-r *REGION*, --region *REGION*

指定 AWS 區域 要使用的 。使用 pcluster configure命令預設為 AWS 區域 指定的 。

-nr, --norollback

停用發生錯誤時的 AWS CloudFormation 堆疊復原。

預設為 False。

-nw, --nowait

表示在處理堆疊命令之後不要等待堆疊事件。

預設為 False。

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

指定叢集範本使用的區段。

-p *EXTRA_PARAMETERS*, --extra-parameters *EXTRA_PARAMETERS*

將額外的參數新增至堆疊更新。

-rd, --reset-desired

將 Auto Scaling 群組的目前容量重設為初始組態值。

預設為 False。

--yes

自動假設所有提示的答案為是。這不應與--force引數結合。

```
$ pcluster update -c path/to/config mycluster
Retrieving configuration from CloudFormation for cluster mycluster...
Validating configuration file .parallelcluster/config...
Found Configuration Changes:

#    parameter                  old value    new value
---  -------------------------  -----------  -----------
```

```
     [compute_resource default]
01   min_count                   1            2
02   max_count                   5            12

Validating configuration update...
Congratulations! The new configuration can be safely applied to your cluster.
Do you want to proceed with the update? - Y/N: Y
Updating: mycluster
Calling update_stack
Status: parallelcluster-mycluster - UPDATE_COMPLETE
```

當命令被呼叫並開始輪詢該呼叫的狀態時，使用「Ctrl-C」結束是安全的。您可以透過呼叫 `pcluster status mycluster` 來返回檢視目前狀態。

pcluster version

顯示 AWS ParallelCluster 版本。

```
pcluster version [ -h ]
```

如需命令特定旗標，請執行：`pcluster [command] --help`。

具名引數

-h, --help

顯示 的說明文字`pcluster version`。

當命令被呼叫並開始輪詢該呼叫的狀態時，使用「Ctrl-C」結束是安全的。您可以透過呼叫 `pcluster status mycluster` 來返回檢視目前狀態。

```
$ pcluster version
2.11.7
```

pcluster-config

更新 AWS ParallelCluster 組態檔案。

```
pcluster-config [ -h ] [convert]
```

如需命令特定旗標，請執行：pcluster-config [command] -h。

具名引數

-h, --help

顯示 的說明文字pcluster-config。

> **Note**
>
> pcluster-config 命令已在 2.9.0 AWS ParallelCluster 版中新增。

子命令

pcluster-config convert

```
pcluster-config convert [ -h ] [ -c CONFIG_FILE ] [ -t CLUSTER_TEMPLATE ]
                        [ -o OUTPUT_FILE ]
```

具名引數

-h, --help

顯示 的說明文字pcluster-config convert。

-c *CONFIG_FILE*, --config-file *CONFIG_FILE*

指定要讀取之組態檔案的路徑。

預設為 ~/.parallelcluster/config。

如需詳細資訊，請參閱設定 AWS ParallelCluster。

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

指示[cluster] 區段要使用的 。如果未指定此引數， pcluster-config convert將使用 中的cluster_template設定[global] 區段。如果未指定，則會使用 [cluster default]區段。

-o *OUTPUT_FILE*, --output *OUTPUT_FILE*

指定要寫入之轉換組態檔案的路徑。根據預設，輸出會寫入 STDOUT。

範例：

```
$ pcluster-config convert -t alpha -o ~/.parallelcluster/multiinstance
```

轉換 [cluster alpha]區段中指定的叢集組態~/.parallelcluster/config，將轉換後的組態檔案寫入 ~/.parallelcluster/multiinstance。

組態

依預設，AWS ParallelCluster會將~/.parallelcluster/config檔案用於所有組態參數。您可以使用或指--config令行選項-c或AWS_PCLUSTER_CONFIG_FILE環境變數來指定自訂組態檔案。

範例組態檔會與 AWS ParallelCluster 檔案一起安裝在 site-packages/aws-parallelcluster/examples/config 的 Python 目錄中。此外GitHub，您也可以在中取得範例組態檔案https://github.com/aws/aws-parallelcluster/blob/v2.11.9/cli/src/pcluster/examples/config。

目前的AWS ParallelCluster第 2 個版本:2.11.9.

主題

配置

AWS ParallelCluster 組態定義在多個區段中。

需要下列[global]區段：剖面和[aws]剖面。

您還必須包括至少一個[cluster]截面和一個[vpc]截面。

區段的開頭為括號括住的區段名稱，接著是參數和組態。

```
[global]
cluster_template = default
update_check = true
sanity_check = true
```

[global] 區段

指定與 pcluster 相關的全域組態選項。

```
[global]
```

主題

- cluster_template
- update_check
- sanity_check

cluster_template

定義預設用於叢集的cluster區段名稱。如需剖cluster面的其他資訊，請參閱〈
一[cluster]節。叢集名稱必須以字母開頭，不得超過 60 個字元，且只能包含字母、數字和連字號
(-)。

例如，以下設定會指定預設使用會啟動 [cluster default] 的區段。

```
cluster_template = default
```

更新政策：更新期間不會分析此設定。

update_check

(選擇性) 檢查的更新pcluster。

預設值為 true。

```
update_check = true
```

更新政策：更新期間不會分析此設定。

sanity_check

(選擇性) 嘗試驗證叢集參數中定義之資源的組態。

預設值為 true。

> **Warning**
>
> 如果設定sanity_check為false，則會略過重要檢查。這可能會導致您的組態無法如預期般運作。

```
sanity_check = true
```

> **Note**
>
> 在AWS ParallelCluster版本 2.5.0 之前，sanity_check預設為. false

更新政策：更新期間不會分析此設定。

[aws] 區段

(選擇性) 用於選取AWS 區域。

叢集建立會使用此優先順序來選AWS 區域取新叢集的：

1. -r或--region參數pcluster create。
2. AWS_DEFAULT_REGION環境變數。
3. aws_region_name在AWS ParallelCluster組態檔案的[aws]區段中設定 (預設位置為~/.parallelcluster/config.) 這是指pcluster configure令更新的位置。

4. region在AWS CLI組態檔案 (~/.aws/config.) [default] 區段中的設定

> **Note**
>
> 在AWS ParallelCluster版本 2.10.0 之前，需要這些設定並套用至所有叢集。

若要存放登入資料，您可以使用環境、Amazon EC2 的 IAM 角色，或使用 AWS CLI，而不是將登入資料儲存到AWS ParallelCluster設定檔中。

```
[aws]
aws_region_name = Region
```

更新政策：更新期間不會分析此設定。

[aliases] 區段

指定別名，並可讓您自訂 ssh 命令。

請注意下列預設設定：

- CFN_USER設定為作業系統的預設使用者名稱
- MASTER_IP被設置為頭節點的 IP 地址
- ARGS被設置為用戶提供的任何參數 *pcluster ssh cluster_name*

```
[aliases]
# This is the aliases section, you can configure
# ssh alias here
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

更新政策：更新期間不會分析此設定。

[cluster] 區段

定義可用於建立叢集的叢集範本。組態檔案可以包含多個[cluster]區段。

相同的叢集範本可用來建立多個叢集。

格式是 [cluster *cluster-template-name*]。[cluster] 依預設會使用 區段中由 cluster_template設定命名的[global]區段，但可以在pcluster命令列上覆寫。

cluster-template-name 必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

```
[cluster default]
```

主題

- additional_cfn_template
- additional_iam_policies
- base_os
- cluster_resource_bucket
- cluster_type
- compute_instance_type
- compute_root_volume_size
- custom_ami
- cw_log_settings
- dashboard_settings
- dcv_settings
- desired_vcpus
- disable_cluster_dns
- disable_hyperthreading
- ebs_settings
- ec2_iam_role
- efs_settings
- enable_efa
- enable_efa_gdr
- enable_intel_hpc_platform
- encrypted_ephemeral
- ephemeral_dir

- extra_json
- fsx_settings
- iam_lambda_role
- initial_queue_size
- key_name
- maintain_initial_size
- master_instance_type
- master_root_volume_size
- max_queue_size
- max_vcpus
- min_vcpus
- placement
- placement_group
- post_install
- post_install_args
- pre_install
- pre_install_args
- proxy_server
- queue_settings
- raid_settings
- s3_read_resource
- s3_read_write_resource
- scaling_settings
- scheduler
- shared_dir
- spot_bid_percentage
- spot_price
- tags
- template_url
- vpc_settings

additional_cfn_template

（選用）定義要與叢集一起啟動的其他 AWS CloudFormation 範本。此額外範本用於建立叢集外的資源，但這些資源是叢集生命週期的一部分。

值必須是公有範本HTTPURL的 ，並提供所有參數。

沒有預設值。

```
additional_cfn_template = https://<bucket-name>.s3.amazonaws.com/my-cfn-template.yaml
```

更新政策：如果變更此設定，則不允許更新。

additional_iam_policies

（選用）指定 Amazon IAM政策的 Amazon Resource Names（ARNs）清單EC2。除了逗號 AWS ParallelCluster 分隔所需的許可之外，此清單還會連接到叢集中使用的根角色。IAM 政策名稱及其ARN不同。名稱無法用作 的引數additional_iam_policies。

如果您的意圖是將額外的政策新增至叢集節點的預設設定，建議您使用 additional_iam_policies設定傳遞額外的自訂IAM政策，而不是使用 ec2_iam_role設定來新增特定EC2政策。這是因為 additional_iam_policies 已新增至 AWS ParallelCluster 所需的預設許可。現有的 ec2_iam_role 必須包含所有必要的許可。但是，由於新增功能時，所需的許可通常會從版本變更為版本，因此現有 ec2_iam_role可能會過時。

沒有預設值。

```
additional_iam_policies = arn:aws:iam::123456789012:policy/CustomEC2Policy
```

> **Note**
>
> additional_iam_policies 2.5.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：此設定可以在更新期間變更。

base_os

(必要) 指定將哪個作業系統類型用於叢集。

可用選項如下：

- alinux2
- centos7
- ubuntu1804
- ubuntu2004

> **Note**
>
> 對於以 AWS Graviton 為基礎的執行個體，僅ubuntu2004支援 ubuntu1804、alinux2或 。

> **Note**
>
> centos8 2.11.4 AWS ParallelCluster 版已移除 的支援。在 ubuntu2004 2.11.0 AWS ParallelCluster 版中已新增 的支援，並ubuntu1604已移除 alinux和 的支援。centos8 已新增 的支援，並在 2.10.0 AWS ParallelCluster 版centos6中移除 的支援。alinux2 2.6.0 版中 AWS ParallelCluster 已新增 的支援。ubuntu1804 已新增 的支援，並在 2.5.0 AWS ParallelCluster 版ubuntu1404中移除 的支援。

除了下表中 AWS 區域 提到的特定不支援 centos7。所有其他 AWS 商業區域都支援下列所有作業系統。

分割區（AWS 區域）	**alinux2**	**centos7**	**ubuntu1804 和 ubuntu2004**
商業（ AWS 區域 未特別提及）	True	True	True
AWS GovCloud（美國東部）（us-gov-east-1 ）	True	False	True
AWS GovCloud（美國西部）（us-gov-west-1 ）	True	False	True
中國 (北京) (cn-north-1 ）	True	False	True

分割區（AWS 區域）	**alinux2**	**centos7**	**ubuntu1804 和 ubuntu2004**
中國 (寧夏) (cn-northwest-1)	True	False	True

> **Note**
>
> base_os 參數也會決定用來登入叢集的使用者名稱。

- centos7: centos
- ubuntu1804 和 ubuntu2004：ubuntu
- alinux2: ec2-user

> **Note**
>
> 在 2.7.0 AWS ParallelCluster 版之前， base_os 參數為選用，預設值為 alinux。從 2.7.0 版開始 AWS ParallelCluster ，需要 base_os 參數。

> **Note**
>
> 如果 scheduler 參數為 awsbatch，則僅支援 alinux2。

```
base_os = alinux2
```

更新政策：如果變更此設定，則不允許更新。

cluster_resource_bucket

（選用）指定用於託管叢集建立時產生的資源的 Amazon S3 儲存貯體名稱。儲存貯體必須已啟用版本控制。如需詳細資訊，請參閱Amazon Simple Storage Service 使用者指南中的使用版本控制。此儲存貯體可用於多個叢集。儲存貯體必須與叢集位於相同的區域。

如果未指定此參數，則會在建立叢集時建立新的儲存貯體。新儲存貯體的名稱為 parallelcluster-*random_string*。在此名稱中，*random_string* 是英數字元的隨機字串。所有叢集資源都存放在具有 表單 的路徑中*bucket_name*/*resource_directory*。resource_directory 具有 表單 *stack_name*-*random_string*，其中 *stack_name* 是 使用的其中一個 AWS CloudFormation 堆疊的名稱 AWS ParallelCluster。的值 *bucket_name* 可以在parallelcluster-*clustername*堆疊輸出中的 ResourcesS3Bucket 值中找到。的值 *resource_directory* 可以在來自相同堆疊的ArtifactS3RootDirectory輸出值中找到。

預設值為 parallelcluster-*random_string*。

```
cluster_resource_bucket = amzn-s3-demo-bucket
```

> **Note**
>
> cluster_resource_bucket 2.10.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：如果變更此設定，則不允許更新。無法強制更新此設定。

cluster_type

（選用）定義要啟動的叢集類型。如果已定義queue_settings設定，則必須將此設定取代為[queue]區段 中的compute_type設定。

有效選項為：ondemand 和 spot。

預設值為 ondemand。

如需 Spot 執行個體的詳細資訊，請參閱 使用 競價型執行個體。

> **Note**
>
> 使用 Spot 執行個體需要AWSServiceRoleForEC2Spot服務連結角色存在於您的帳戶中。若要使用 在帳戶中建立此角色 AWS CLI，請執行下列命令：
>
> ```
> aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
> ```
>
> 如需詳細資訊，請參閱 Amazon EC2使用者指南 中的 Spot 執行個體請求的服務連結角色。

```
cluster_type = ondemand
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

compute_instance_type

（選用）定義用於叢集運算節點的 Amazon EC2執行個體類型。執行個體類型的架構必須與master_instance_type用於設定的架構相同。如果已定義queue_settings設定，則必須將此設定取代為[compute_resource]區段 中的instance_type設定。

如果您使用的是awsbatch排程器，請參閱 AWS Batch 使用者介面中的運算環境建立，以取得支援的執行個體類型清單。

當排程器為 awsbatch 時，預設為 t2.micro、optimal。

```
compute_instance_type = t2.micro
```

> **Note**
>
> 2.8.0 AWS ParallelCluster 版中已新增對 AWS Graviton 型執行個體（包括 A1和 C6g執行個體）的支援。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

compute_root_volume_size

（選用）以 GB（GiB）為單位指定 ComputeFleet 根磁碟區大小。必須AMI支援 growroot。

預設值為 35。

> **Note**
>
> 對於 2.5.0 和 2.10.4 之間的 AWS ParallelCluster 版本，預設值為 25。在 2.5.0 AWS ParallelCluster 版之前，預設值為 20。

```
compute_root_volume_size = 35
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

custom_ami

（選用）指定AMI要用於主機和運算節點的自訂 ID，而不是預設發佈的 AMIs。如需詳細資訊，請參閱修改 AMI 或 建置自訂 AWS ParallelCluster AMI。

沒有預設值。

```
custom_ami = ami-00d4efc81188687a0
```

如果自訂AMI需要額外的許可才能啟動，則必須將這些許可新增至使用者和前端節點政策。

例如，如果自訂AMI具有與其相關聯的加密快照，則使用者和頭節點政策都需要下列其他政策：

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "kms:DescribeKey",
               "kms:ReEncrypt*",
               "kms:CreateGrant",
               "kms:Decrypt"
           ],
           "Resource": [
               "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
           ]
       }
   ]
}
```

更新政策：如果變更此設定，則不允許更新。

cw_log_settings

（選用）使用 CloudWatch Logs 組態識別 [cw_log]區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱[cw_log]章節、 Amazon CloudWatch 儀表板和 與 Amazon CloudWatch Logs 整合。

例如，下列設定指定開始的區段[cw_log custom-cw]用於 CloudWatch Logs 組態。

```
cw_log_settings = custom-cw
```

> **Note**
>
> cw_log_settings 2.6.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：如果變更此設定，則不允許更新。

dashboard_settings

（選用）使用 CloudWatch儀表板組態識別 [dashboard]區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱[dashboard]一節。

例如，下列設定指定開始的區段[dashboard custom-dashboard用於 CloudWatch 儀表板組態。

```
dashboard_settings = custom-dashboard
```

> **Note**
>
> dashboard_settings 2.10.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：此設定可以在更新期間變更。

dcv_settings

（選用）使用 Amazon DCV組態識別 [dcv]區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱 [dcv]一節。

例如，下列設定會指定 [dcv custom-dcv] Amazon DCV組態使用的啟動區段。

```
dcv_settings = custom-dcv
```

> **Note**
>
> 在 AWS Graviton 型執行個體上，Amazon DCV僅支援 `alinux2`。

> **Note**
>
> `dcv_settings` 2.5.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：如果變更此設定，則不允許更新。

desired_vcpus

（選用）在 vCPUs 運算環境中指定所需的 數目。僅在排程器是 `awsbatch` 時才使用。

預設值為 4。

```
desired_vcpus = 4
```

更新政策：更新期間不會分析此設定。

disable_cluster_dns

（選用）指定是否不應建立叢集DNS的項目。根據預設， 會 AWS ParallelCluster 建立 Route 53 託管區域。如果 `disable_cluster_dns` 設定為 `true`，則不會建立託管區域。

預設值為 `false`。

```
disable_cluster_dns = true
```

> **Warning**
>
> 叢集需要名稱解析系統才能正常運作。如果 `disable_cluster_dns` 設定為 `true`，則必須提供額外的名稱解析系統。

> **Important**
>
> disable_cluster_dns = true只有在指定queue_settings設定時才支援。

> **Note**
>
> disable_cluster_dns 2.9.1 AWS ParallelCluster 版中已新增 的支援。

更新政策：如果變更此設定，則不允許更新。

disable_hyperthreading

（選用）停用主機和運算節點上的超執行緒。並非所有執行個體類型都可以停用超執行緒。如需支援停用超執行緒的執行個體類型清單，請參閱 Amazon EC2使用者指南 中的CPU每個執行個體類型的每個CPU核心和執行緒。如果已定義queue_settings設定，則可以定義此設定，也可以定義[queue]區段中的disable_hyperthreading設定。

預設值為 false。

```
disable_hyperthreading = true
```

> **Note**
>
> disable_hyperthreading 只會在 時影響主機節點scheduler = awsbatch。

> **Note**
>
> disable_hyperthreading 2.5.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：如果變更此設定，則不允許更新。

ebs_settings

（選用）使用安裝在主機節點上的 Amazon EBS磁碟區來識別[ebs]區段。使用多個 Amazon EBS磁碟區時，請在清單中輸入這些參數，每個參數都以逗號分隔。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

最多支援五（5）個額外的 Amazon EBS磁碟區。

如需詳細資訊，請參閱 [ebs]一節。

例如，下列設定指定 Amazon EBS磁碟區啟動[ebs custom1]和[ebs custom2]使用的區段。

```
ebs_settings = custom1, custom2
```

更新政策：如果變更此設定，則不允許更新。

ec2_iam_role

（選用）定義EC2連接至叢集中所有執行個體的 Amazon 現有IAM角色的名稱。IAM 角色名稱及其 Amazon Resource Name（ARN）不同。ARNs 無法用作 的引數ec2_iam_role。

如果指定此選項，則會忽略 additional_iam_policies 設定。如果您的意圖是將額外的政策新增至叢集節點的預設設定，建議您使用 additional_iam_policies設定傳遞額外的自訂IAM政策，而不是使用 ec2_iam_role設定。

如果未指定此選項，EC2則會使用 Amazon 的預設 AWS ParallelCluster IAM角色。如需詳細資訊，請參閱AWS Identity and Access Management 中的 角色 AWS ParallelCluster。

沒有預設值。

```
ec2_iam_role = ParallelClusterInstanceRole
```

更新政策：如果變更此設定，則不允許更新。

efs_settings

（選用）指定與 Amazon EFS 檔案系統相關的設定。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱[efs]一節。

例如，下列設定指定 [efs customfs] Amazon EFS 檔案系統組態使用啟動的 區段。

```
efs_settings = customfs
```

更新政策：如果變更此設定，則不允許更新。

enable_efa

（選用）如果存在，請指定已為運算節點啟用 Elastic Fabric Adapter（EFA）。若要檢視支援 的EC2執行個體清單EFA，請參閱 Amazon EC2 適用於 Linux 執行個體的使用者指南 中的支援執行個體類型。如需詳細資訊，請參閱Elastic Fabric Adapter。如果已定義queue_settings設定，則可以定義此設定，也可以定義[queue]區段中的enable_efa設定。應使用叢集置放群組以充分減少執行個體之間的延遲。如需詳細資訊，請參閱 placement 和 placement_group。

```
enable_efa = compute
```

> **Note**
>
> 在 2.10.1 AWS ParallelCluster 版中新增了對 EFA Arm 型 Graviton2 執行個體的支援。

更新政策：如果變更此設定，則不允許更新。

enable_efa_gdr

（選用）從 2.11.3 AWS ParallelCluster 版開始，此設定沒有效果。如果執行個體類型和作業系統都支援 GPUDirectRDMA（遠端直接記憶體存取EFA），則一律會啟用 Elastic Fabric Adapter（ ）支援。

> **Note**
>
> AWS ParallelCluster 2.10.0 至 2.11.2 版：如果 compute，則指定為運算節點啟用 Elastic Fabric Adapter（EFA）支援 GPUDirectRDMA（遠端直接記憶體存取）。將此設定設為 compute需要將enable_efa設定設為 compute。EFA 支援GPUDirectRDMA特定作業系統（ 為 alinux2、centos7、 ubuntu1804或 p4d.24xlarge ）上的特定執行個體類型（base_os）ubuntu2004。如果已定義queue_settings設定，則可以定義此設定，也可以定義[queue]區段中的enable_efa_gdr設定。應使用叢集置放群組以充分減少執行個體之間的延遲。如需詳細資訊，請參閱 placement 和 placement_group。

```
enable_efa_gdr = compute
```

> **Note**
>
> `enable_efa_gdr` 2.10.0 AWS ParallelCluster 版已新增 的支援。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

enable_intel_hpc_platform

（選用）如果存在，則表示接受 Intel Parallel Studio 的最終使用者授權合約。這會導致 Intel Parallel Studio 安裝在主機節點上並與運算節點共用。這會將幾分鐘新增至主機節點開機所需的時間。`enable_intel_hpc_platform` 設定僅在 上受支援 CentOS 7（`base_os = centos7`）。

預設值為 `false`。

```
enable_intel_hpc_platform = true
```

> **Note**
>
> `enable_intel_hpc_platform` 參數與以 AWS Graviton 為基礎的執行個體不相容。

> **Note**
>
> `enable_intel_hpc_platform` 2.5.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：如果變更此設定，則不允許更新。

encrypted_ephemeral

（選用）使用 LUKS（Linux 統一金鑰設定），使用不可復原的記憶體內金鑰加密暫時性執行個體存放磁碟區。

如需詳細資訊，請參閱https://gitlab.com/cryptsetup/cryptsetup/blob/master/README.md。

預設值為 `false`。

```
encrypted_ephemeral = true
```

更新政策：如果變更此設定，則不允許更新。

ephemeral_dir

（選用）定義在使用執行個體存放區磁碟區時掛載它們的路徑。

預設值為 /scratch。

```
ephemeral_dir = /scratch
```

更新政策：如果變更此設定，則不允許更新。

extra_json

（選用）定義合併到 JSON的額外項目 Chef dna.json。如需詳細資訊，請參閱 建置自訂 AWS ParallelCluster AMI。

預設值為 {}。

```
extra_json = {}
```

> **Note**
>
> 從 2.6.1 AWS ParallelCluster 版開始，在啟動節點時，依預設會略過大部分的安裝配方，以改善啟動時間。若要以啟動時間的費用執行所有安裝配方，以獲得更好的向後相容性，請將 "skip_install_recipes" : "no"新增至 extra_json 設定中的 cluster 金鑰。例如：
>
> ```
> extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
> ```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

fsx_settings

（選用）指定定義 FSx Lustre 組態的 區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱 [fsx]一節。

例如，下列設定指定啟動的 區段[fsx fs]用於 FSx for Lustre 組態。

```
fsx_settings = fs
```

更新政策：如果變更此設定，則不允許更新。

iam_lambda_role

（選用）定義現有 AWS Lambda 執行角色的名稱。此角色會連接至叢集中的所有 Lambda 函數。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 執行角色。

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

IAM 角色名稱及其 Amazon Resource Name（ARN）不同。ARNs 無法用作 的引數iam_lambda_role。如果同時定義 iam_lambda_role ec2_iam_role和 ，且 scheduler為 sge、 slurm或 torque，則不會建立任何角色。如果 scheduler是 awsbatch，則會在 期間建立角色pcluster start。如需政策範例，請參閱 ParallelClusterLambdaPolicy 使用 SGE, Slurm，或 Torque和 ParallelClusterLambdaPolicy 使用 awsbatch。

沒有預設值。

```
iam_lambda_role = ParallelClusterLambdaRole
```

> **Note**
>
> iam_lambda_role 2.10.1 AWS ParallelCluster 版中已新增 的支援。

更新政策：此設定可以在更新期間變更。

initial_queue_size

（選用）將 Amazon EC2執行個體的初始數量設定為在叢集中啟動的運算節點。如果已定義queue_settings設定，則必須移除此設定，並由[compute_resource]區段 中的initial_count設定取代。

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

此設定僅適用於傳統排程器（SGE, Slurm 和 Torque）。 如果maintain_initial_size設定為 true，則initial_queue_size設定必須至少為一（1）個。

如果排程器是 awsbatch，請改用 min_vcpus。

預設為 2。

```
initial_queue_size = 2
```

更新政策：此設定可以在更新期間變更。

key_name

（選用）為現有的 Amazon EC2金鑰對命名，以啟用對執行個體的SSH存取。

```
key_name = mykey
```

> **Note**
>
> 在 2.11.0 AWS ParallelCluster 版之前， key_name是必要的設定。

更新政策：如果變更此設定，則不允許更新。

maintain_initial_size

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

（選用）維護傳統排程器 Auto Scaling 群組的初始大小（SGE, Slurm 和 Torque).

如果排程器是 awsbatch，請改用 desired_vcpus。

此設定是布林值旗標。如果設定為 `true`，Auto Scaling 群組的成員不會少於 的值initial_queue_size，且 的值initial_queue_size必須是一（1）或更高。叢集仍可擴充至 max_queue_size 值。如果 `cluster_type = spot` ，Auto Scaling 群組可以中斷執行個體，且大小可能會降至 以下initial_queue_size。

如果設定為 `false`，Auto Scaling 群組可以縮減為零（0）成員，以防止資源在不需要時閒置。

如果已定義 queue_settings 設定，則必須移除此設定，並由[compute_resource]區段 中的 initial_count和 min_count設定取代。

預設為 `false`。

```
maintain_initial_size = false
```

更新政策：此設定可以在更新期間變更。

master_instance_type

（選用）定義用於主機節點的 Amazon EC2執行個體類型。執行個體類型的架構必須與用於compute_instance_type設定的架構相同。

在具有 免費方案 AWS 區域 的 中，預設為 免費方案執行個體類型（`t2.micro` 或 `t3.micro`）。在中 AWS 區域 沒有免費方案的 中，預設為 `t3.micro`。如需 AWS 免費方案的詳細資訊，請參閱 AWS 免費方案 FAQs。

```
master_instance_type = t2.micro
```

> **Note**
>
> 在 2.10.1 AWS ParallelCluster 版之前， 在所有 `t2.micro`中預設為 AWS 區域。在 2.10.0 AWS ParallelCluster 版中， `p4d.24xlarge`不支援主機節點。2.8.0 AWS ParallelCluster 版中已新增對 AWS Graviton 型執行個體（例如 `A1`和 `C6g`）的支援。

更新政策：如果變更此設定，則不允許更新。

master_root_volume_size

（選用）以 GB（GiB）為單位指定主機節點根磁碟區大小。必須AMI支援 `growroot`。

預設值為 35。

> **Note**
>
> 對於 2.5.0 和 2.10.4 之間的 AWS ParallelCluster 版本，預設值為 25。在 2.5.0 AWS ParallelCluster 版之前，預設值為 20。

```
master_root_volume_size = 35
```

更新政策：如果變更此設定，則不允許更新。

max_queue_size

（選用）設定可在叢集中啟動的 Amazon EC2執行個體數量上限。如果已定義queue_settings設定，則必須移除此設定，並以[compute_resource]區段 中的max_count設定取代。

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

此設定僅適用於傳統排程器（SGE, Slurm 和 Torque).

如果排程器是 awsbatch，請改用 max_vcpus。

預設為 10。

```
max_queue_size = 10
```

更新政策：此設定可以在更新期間變更，但如果值減少，則應停止運算機群。否則，現有的節點可能會終止。

max_vcpus

（選用）指定 vCPUs 運算環境中的 數目上限。僅在排程器是 awsbatch 時才使用。

預設值為 20。

```
max_vcpus = 20
```

更新政策：此設定無法在更新期間減少。

min_vcpus

（選用）維護awsbatch排程器 Auto Scaling 群組的初始大小。

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

如果排程器是 SGE, Slurm，或 Torque，請maintain_initial_size改用 。

運算環境的成員數不得少於 min_vcpus 的值。

預設為 0。

```
min_vcpus = 0
```

更新政策：此設定可以在更新期間變更。

placement

（選用）定義叢集置放群組邏輯，讓整個叢集或僅運算執行個體可以使用叢集置放群組。

如果已定義queue_settings設定，則應移除此設定，並以每個區段placement_group的設定取代。 [queue]如果相同的置放群組用於不同的執行個體類型，則請求可能會因為容量不足錯誤而失敗。如需詳細資訊，請參閱 Amazon EC2使用者指南 中的執行個體容量不足。多個佇列只有在預先建立並在每個佇列placement_group的設定中設定時，才能共用置放群組。如果每個[queue]區段都定義了placement_group設定，則主機節點不能位於佇列的置放群組中。

有效選項為 cluster 或 compute。

當排程器為 時，不會使用此參數awsbatch。

預設值為 compute。

```
placement = compute
```

更新政策：如果變更此設定，則不允許更新。

placement_group

（選用）定義叢集置放群組。如果已定義queue_settings設定，則應移除此設定，並由[queue]區段 中的placement_group設定取代。

有效選項為下列值：

- DYNAMIC
- 現有的 Amazon EC2叢集置放群組名稱

設為 DYNAMIC 時，即會建立唯一的置放群組，並在叢集堆疊過程將其刪除。

當排程器為 時，不會使用此參數awsbatch。

如需置放群組的詳細資訊，請參閱 Amazon EC2使用者指南 中的置放群組。如果相同的置放群組用於不同的執行個體類型，則請求可能會因為容量不足錯誤而失敗。如需詳細資訊，請參閱 Amazon EC2使用者指南 中的執行個體容量不足。

沒有預設值。

並非所有執行個體類型都支援叢集置放群組。例如， 的預設執行個體類型t3.micro不支援叢集置放群組。如需支援叢集置放群組的執行個體類型清單的相關資訊，請參閱 Amazon EC2使用者指南 中的叢集置放群組規則和限制。如需使用置放群組的秘訣，請參閱 置放群組和執行個體啟動問題。

```
placement_group = DYNAMIC
```

更新政策：如果變更此設定，則不允許更新。

post_install

（選用）指定在所有節點引導動作完成後執行URL的安裝後指令碼的 。如需詳細資訊，請參閱自訂引導操作。

使用 awsbatch做為排程器時，安裝後指令碼只會在主機節點上執行。

參數格式可以是 http://*hostname/path/to/script.sh* 或 s3://*bucketname/path/to/script.sh*。

沒有預設值。

```
post_install = s3://<bucket-name>/my-post-install-script.sh
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

post_install_args

（選用）指定要傳遞至安裝後指令碼的引數引號清單。

沒有預設值。

```
post_install_args = "argument-1 argument-2"
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

pre_install

（選用）指定在啟動任何節點部署引導動作之前執行URL的預先安裝指令碼的 。如需詳細資訊，請參閱自訂引導操作。

使用 awsbatch做為排程器時，預先安裝指令碼只會在主機節點上執行。

參數格式可以是 http://*hostname/path/to/script.sh* 或 s3://*bucketname/path/to/script.sh*。

沒有預設值。

```
pre_install = s3://<bucket-name>/my-pre-install-script.sh
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

pre_install_args

（選用）指定要傳遞至預先安裝指令碼的引數引號清單。

沒有預設值。

```
pre_install_args = "argument-3 argument-4"
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

proxy_server

（選用）定義 HTTP或 HTTPS Proxy 伺服器，通常為 `http://x.x.x.x:8080`。

沒有預設值。

```
proxy_server = http://10.11.12.13:8080
```

更新政策：如果變更此設定，則不允許更新。

queue_settings

（選用）指定叢集使用佇列而非同質運算機群，以及使用哪些[queue]區段。列出的第一個[queue]區段是預設排程器佇列。queue 區段名稱必須以小寫字母開頭，包含不超過 30 個字元，且僅包含小寫字母、數字和連字號（-）。

> **Important**
>
> queue_settings 只有在 scheduler 設定為 時受支援slurm。不得指定 cluster_type、compute_instance_type、initial_queue_size、maintain_initial_siz placement_group和 spot_price設定。disable_hyperthreading 和 enable_efa設定可以在 [cluster]區段或 [queue] 區段中指定，但不能同時指定兩者。

最多支援五（5）個[queue]區段。

如需詳細資訊，請參閱[queue]一節。

例如，下列設定指定啟動[queue q1]和[queue q2]使用的區段。

```
queue_settings = q1, q2
```

> **Note**
>
> queue_settings 2.9.0 AWS ParallelCluster 版中已新增 的支援。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

raid_settings

（選用）使用 Amazon EBS磁碟區RAID組態識別 [raid]區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱 [raid]一節。

例如，下列設定指定[raid rs]開始用於 Auto Scaling 組態的區段。

```
raid_settings = rs
```

更新政策：如果變更此設定，則不允許更新。

s3_read_resource

（選用）指定 Amazon S3 資源，其中 AWS ParallelCluster 節點被授予唯讀存取權。

例如，arn:aws:s3:::*my_corporate_bucket*提供對 的唯讀存取權 *my_corporate_bucket* 儲存貯體和 至儲存貯體中的物件。

如需格式的詳細資訊，請參閱使用 Amazon S3。

沒有預設值。

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket*
```

更新政策：此設定可以在更新期間變更。

s3_read_write_resource

（選用）指定 Amazon S3 資源，哪些 AWS ParallelCluster 節點被授予讀取/寫入存取權。

例如，arn:aws:s3:::*my_corporate_bucket*/Development/* 提供對 Development資料夾中所有物件的讀取/寫入存取權 *my_corporate_bucket* 儲存貯體。

如需格式的詳細資訊，請參閱使用 Amazon S3。

沒有預設值。

```
s3_read_write_resource = arn:aws:s3:::my_corporate_bucket/*
```

更新政策：此設定可以在更新期間變更。

scaling_settings

使用 Auto Scaling 組態識別 [scaling]區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱 [scaling]一節。

例如，下列設定指定 Auto Scaling 組態[scaling custom]使用啟動的區段。

```
scaling_settings = custom
```

更新政策：如果變更此設定，則不允許更新。

scheduler

(必要) 定義叢集的排程器。

有效選項為下列值：

awsbatch

AWS Batch

如需awsbatch排程器的詳細資訊，請參閱聯網設定 和 AWS Batch (awsbatch)。

sge

> **Note**
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

Son of Grid Engine (SGE)

slurm

Slurm Workload Manager (Slurm)

`torque`

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

Torque Resource Manager (Torque)

> **Note**
>
> 在 2.7.0 AWS ParallelCluster 版之前， `scheduler` 參數為選用，預設值為 `sge`。從 2.7.0 AWS ParallelCluster 版開始，需要 `scheduler` 參數。

```
scheduler = slurm
```

更新政策：如果變更此設定，則不允許更新。

shared_dir

（選用）定義掛載共用 Amazon EBS磁碟區的路徑。

請勿將此選項與多個 Amazon EBS磁碟區搭配使用。相反地，請在每個[ebs]區段 下提供shared_dir值。

如需使用多個 Amazon EBS磁碟區的詳細資訊，請參閱[ebs]一節。

預設值為 `/shared`。

下列範例顯示掛載在 的共用 Amazon EBS磁碟區`/myshared`。

```
shared_dir = myshared
```

更新政策：如果變更此設定，則不允許更新。

spot_bid_percentage

（選用）當 `awsbatch`是排程器時 ComputeFleet，設定用於計算 最高 Spot 價格的隨需百分比。

如果未指定，則會選取目前的 Spot 市價，上限為隨需價格。

```
spot_bid_percentage = 85
```

更新政策：此設定可以在更新期間變更。

spot_price

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

（選用）為傳統排程器 ComputeFleet 上的 設定 Spot 價格上限（SGE, Slurm 和 Torque）。 僅在cluster_type設定設為 時使用spot。如果您未指定值，則會向您收取 Spot 價格，以隨需價格為上限。如果已定義queue_settings設定，則必須移除此設定，並由[compute_resource]區段 中的spot_price設定取代。

如果排程器是 awsbatch，請改用 spot_bid_percentage。

如需尋找符合您需求的 Spot 執行個體的協助，請參閱 Spot 執行個體顧問 。

```
spot_price = 1.50
```

> **Note**
>
> 在 2.5.0 AWS ParallelCluster 版中，如果spot_price未指定 cluster_type = spot ，則執行個體會啟動 ComputeFleet 失敗。這是在 2.5.1 AWS ParallelCluster 版中修正的。

更新政策：此設定可以在更新期間變更。

tags

（選用）定義 要使用的標籤 AWS CloudFormation。

如果透過 --tags 指定命令列標籤，則會使用組態標籤來合併它們。

命令列標籤會覆寫具有相同金鑰的組態標籤。

標籤JSON已格式化。請勿在捲曲支架之外使用引號。

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南 中的AWS CloudFormation 資源標籤類型。

```
tags = {"key" : "value", "key2" : "value2"}
```

更新政策：如果變更此設定，則不允許更新。

> **Note**
>
> 更新政策不支援將 2.8.0 版tags的設定 AWS ParallelCluster 變更為 2.9.1 版。
> 對於 2.10.0 版到 2.11.7 版，支援變更tags設定的列出的更新政策並不準確。不支援修改此設定時的叢集更新。

template_url

（選用）定義用來建立叢集的 AWS CloudFormation 範本路徑。

更新使用原先建立堆疊所用的範本。

預設為 https://*aws_region_name*-aws-parallelcluster.s3.amazonaws.com/templates/aws-parallelcluster-*version*.cfn.json。

> **Warning**
>
> 這是進階參數。此設定的任何變更都會由您自行承擔風險。

```
template_url = https://us-east-1-aws-parallelcluster.s3.amazonaws.com/templates/aws-parallelcluster-2.11.9.cfn.json
```

更新政策：更新期間不會分析此設定。

vpc_settings

（必要）使用部署叢集的 Amazon VPC組態來識別 [vpc]區段。區段名稱必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

如需詳細資訊，請參閱 [vpc]一節。

例如，下列設定會指定 [vpc public] Amazon VPC組態使用的啟動區段。

```
vpc_settings = public
```

更新政策：如果變更此設定，則不允許更新。

[compute_resource] 區段

定義計算資源的組態設定。[compute_resource]區段中的compute_resource_settings設定參考[queue]剖面。[compute_resource]僅當設定為時scheduler，才支援剖面slurm。

格式是 [compute_resource *<compute-resource-name>*]。*compute-resource-name*必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[compute_resource cr1]
instance_type = c5.xlarge
min_count = 0
initial_count = 2
max_count = 10
spot_price = 0.5
```

> **Note**
>
> 在AWS ParallelCluster版本 2.9.0 中添加了對該[compute_resource]部分的支持。

主題

- initial_count
- instance_type
- max_count
- min_count
- spot_price

initial_count

(選擇性) 設定要為此運算資源啟動的 Amazon EC2 執行個體的初始數量。至少在計算資源中啟動了這麼多節點之前，叢集建立才會完成。如果佇列的compute_type設定為，spot且沒有足夠的 Spot 執行個體可用，則叢集建立可能會逾時並失敗。任何大於min_count設定的計數均為動態容量，視設定而scaledown_idletime定。此設定會取代 initial_queue_size 設定。

預設為 0。

```
initial_count = 2
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

instance_type

(必要) 定義用於此運算資源的 Amazon EC2 執行個體類型。執行個體類型的架構必須與master_instance_type設定所使用的架構相同。對於截面所參照的每個[compute_resource]截面，此instance_type設定必須是唯一[queue]的。此設定會取代 compute_instance_type 設定。

```
instance_type = t2.micro
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

max_count

(選擇性) 設定可在此運算資源中啟動的 Amazon EC2 執行個體數目上限。任何大於initial_count設定的計數都會以關閉電源模式啟動。此設定會取代 max_queue_size 設定。

預設為 10。

```
max_count = 10
```

更新政策：將佇列的大小減少到目前節點數量以下，需要先停止運算機群。

> **Note**
>
> 在 2.0.0 版到 2.9.1 AWS ParallelCluster 版的計算叢集停止之前，更新原則不支援變更max_count設定。

min_count

(選擇性) 設定可在此運算資源中啟動的 Amazon EC2 執行個體數目下限。這些節點都是靜態容量。在計算資源中啟動至少這個數目的節點之前，叢集建立才會完成。

預設為 0。

```
min_count = 1
```

更新政策：減少佇列中的靜態節點數量需要先停止運算機群。

> **Note**
>
> 在 2.0.0 版到 2.9.1 AWS ParallelCluster 版的計算叢集停止之前，更新原則不支援變更min_count設定。

spot_price

(選擇性) 設定此計算資源的最高 Spot 價格。僅在包含此計算資源之佇列的compute_type設定設為時使用spot。此設定會取代 spot_price 設定。

如果未指定值，系統會以隨需價格上限向您收取 Spot 價格的費用。

如需尋找符合您需求的 Spot 執行個體的協助，請參閱 Spot 執行個體建議程式。

```
spot_price = 1.50
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

[cw_log] 區段

定義CloudWatch記錄檔的組態設定。

格式是 [cw_log *cw-log-name*]。*cw-log-name*必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[cw_log custom-cw-log]
enable = true
retention_days = 14
```

如需詳細資訊，請參閱 與 Amazon CloudWatch Logs 整合、Amazon CloudWatch 儀表板 及 與 Amazon CloudWatch Logs 整合。

> **Note**
>
> 在 2.6.0 AWS ParallelCluster 版本中添加了支持。`cw_log`

enable

(選擇性) 指出CloudWatch記錄檔是否已啟用。

預設值為 `true`。用`false`於停用CloudWatch記錄檔。

下列範例會啟用CloudWatch記錄檔。

```
enable = true
```

更新政策：如果變更此設定，則不允許更新。

retention_days

(選擇性) 指出「記錄檔」會保留個別CloudWatch記錄事件的天數。

預設值為 14。支援的值分別為
1、3、5、7、14、30、60、90、120、150、180、365、400、545、731、1827，和 3653。

下列範例會將記錄檔設定為將CloudWatch記錄事件保留 30 天。

```
retention_days = 30
```

更新政策：此設定可以在更新期間變更。

[dashboard] 區段

定義CloudWatch儀表板的組態設定。

格式為[dashboard *dashboard-name*]。 *#####*必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[dashboard custom-dashboard]
enable = true
```

> **Note**
>
> 在AWS ParallelCluster版本 2.10.0 中添加dashboard了對的支持。

enable

(選擇性) 指出CloudWatch儀表板是否已啟用。

預設值為 true。用false於停用CloudWatch儀表板。

下列範例會啟用CloudWatch儀表板。

```
enable = true
```

更新政策：此設定可以在更新期間變更。

[dcv] 區段

定義在主機節點上執行的 Amazon DCV伺服器的組態設定。

若要建立和設定 Amazon DCV 伺服器，請使用您在 dcv區段中定義的dcv_settings名稱指定叢集，並將 enable設定為 master、 base_os 和 alinux2、 centos7ubuntu1804或 ubuntu2004。如果主機節點是ARM執行個體，請將 base_os設定為 alinux2、 centos7或 ubuntu1804。

格式是 [dcv *dcv-name*]。*dcv-name* 必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

```
[dcv custom-dcv]
enable = master
port = 8443
access_from = 0.0.0.0/0
```

如需詳細資訊，請參閱 透過 Amazon 連線至主機節點 DCV

> **Important**
>
> 根據預設， 的 Amazon DCV 連接埠設定 AWS ParallelCluster 會開放給所有IPv4地址。不過，只有在您擁有適用於 Amazon DCV工作階段URL的 時，才能連線至 Amazon DCV 連接埠，並在從 URL傳回 後 30 秒內連線至 Amazon DCV工作階段pcluster dcv connect。使

用 access_from設定進一步限制對 IP 範圍為 CIDR的 Amazon DCV 連接埠的存取，並使用 port設定來設定非標準連接埠。

> **Note**
>
> centos8 2.10.4 版已 AWS ParallelCluster 移除 上的 [dcv]區段支援。2.10.0 AWS ParallelCluster 版中centos8已新增 上的 [dcv]章節支援。在 2.9.0 AWS ParallelCluster 版中新增了對 AWS Graviton 型執行個體[dcv]一節的支援。2.6.0 AWS ParallelCluster 版已ubuntu1804新增 alinux2和 的 [dcv]章節支援。2.5.0 AWS ParallelCluster 版中centos7已新增 上的 [dcv]章節支援。

access_from

（選用，建議）指定 Amazon 連線的 CIDR格式化 IP 範圍DCV。此設定只會在 AWS ParallelCluster 建立安全群組時使用。

預設值是 0.0.0.0/0，該值允許從任何網際網路位址存取。

```
access_from = 0.0.0.0/0
```

更新政策：此設定可以在更新期間變更。

enable

（必要）指示是否DCV在主機節點上啟用 Amazon。若要在主機節點DCV上啟用 Amazon 並設定所需的安全群組規則，請將 enable設定設為 master。

下列範例會在主機節點DCV上啟用 Amazon。

```
enable = master
```

> **Note**
>
> Amazon DCV會自動產生自我簽署憑證，用於保護在主機節點上執行的 Amazon DCV用戶端與 Amazon DCV 伺服器之間的流量。若要設定您自己的憑證，請參閱Amazon DCVHTTPS憑證。

更新政策：如果變更此設定，則不允許更新。

port

（選用）指定 Amazon 的連接埠DCV。

預設值為 8443。

```
port = 8443
```

更新政策：如果變更此設定，則不允許更新。

[ebs] 區段

針對掛接在頭節點上並透過 NFS 共用至運算節點的磁碟區，定義 Amazon EBS 磁碟區組態設定。

若要了解如何在叢集定義中包含 Amazon EBS 磁碟區，請參閱[cluster] ##/ebs_settings。

若要將現有的 Amazon EBS 磁碟區用於獨立於叢集生命週期的長期永久儲存，請指定ebs_volume_id。

如果未指定ebs_volume_id，請在 AWS ParallelCluster 建立叢集時從[ebs]設定建立 EBS 磁碟區建立 EBS 磁碟區，並在刪除叢集時刪除磁碟區和資料。

如需詳細資訊，請參閱 最佳做法：將叢集移至新叢集 AWS ParallelCluster 次要或修補程式版本。

格式為[ebs *ebs-name*]。 *ebs-name* 必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[ebs custom1]
shared_dir = vol1
ebs_snapshot_id = snap-xxxxx
volume_type = io1
volume_iops = 200
...

[ebs custom2]
shared_dir = vol2
...
```

```
...
```

主題

- shared_dir
- ebs_kms_key_id
- ebs_snapshot_id
- ebs_volume_id
- encrypted
- volume_iops
- volume_size
- volume_throughput
- volume_type

shared_dir

(必要) 指定掛接共用 Amazon EBS 磁碟區的路徑。

使用多個 Amazon EBS 磁碟區時，需要使用此參數。

當您使用一個 Amazon EBS 磁碟區時，此選項會覆寫區段下指shared_dir定的磁碟區。[cluster]在下列範例中，磁碟區會掛載至 /vol1。

```
shared_dir = vol1
```

更新政策：如果變更此設定，則不允許更新。

ebs_kms_key_id

(選擇性) 指定用於加密的自訂 AWS KMS 金鑰。

此參數必須與 encrypted = true 搭配使用。它還必須具有自訂 ec2_iam_role。

如需詳細資訊，請參閱 使用自訂 KMS 金鑰進行磁碟加密。

```
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

更新政策：如果變更此設定，則不允許更新。

ebs_snapshot_id

(選擇性) 如果您使用快照作為磁碟區的來源，請定義 Amazon EBS 快照 ID。

沒有預設值。

```
ebs_snapshot_id = snap-xxxxx
```

更新政策：如果變更此設定，則不允許更新。

ebs_volume_id

(選擇性) 定義要連接至主節點的現有 Amazon EBS 磁碟區的磁碟區識別碼。

沒有預設值。

```
ebs_volume_id = vol-xxxxxx
```

更新政策：如果變更此設定，則不允許更新。

encrypted

(選擇性) 指定 Amazon EBS 磁碟區是否已加密。注意：請不要與快照搭配使用。

預設值為 `false`。

```
encrypted = false
```

更新政策：如果變更此設定，則不允許更新。

volume_iops

(選擇性) 定義`io1`、`io2`和`gp3`類型磁碟區的 IOPS 數目。

預設值、支援的值和總`volume_iops`成`volume_size`比例會因`volume_type`和而異`volume_size`。

volume_type = io1

預設值 volume_iops = 100

支援的值 volume_iops = 100 至 64 萬 †

最大volume_iops比volume_size率 = 每個 GiB 的 50 IOPS。5000 IOPS 需要一volume_size個至少 100 GiB。

volume_type = io2

預設值 volume_iops = 100

支援的值 volume_iops = 100 至 64 萬 (對於io2區塊快速磁碟區為 256000) †

每volume_iops個 GiB 的最大volume_size比例 = 500 IOPS。5000 IOPS 需要至少 10 GiB博volume_size的一個。

volume_type = gp3

預設volume_iops值

支援的volume_iops值：

每volume_iops個 GiB 的最大volume_size比例 = 500 IOPS。5000 IOPS 需要至少 10 GiB博volume_size的一個。

```
volume_iops = 200
```

更新政策：此設定可以在更新期間變更。

† 只有在已佈建超過 32,000 IOPS 的硝基系統上建置的執行個體上才能保證最大 IOPS。其他執行個體可保證高達 32,000 IOPS。除非您修改磁碟區，否則舊版io1磁碟區可能無法達到完整效能。 io2區塊快速磁碟區支援的volume_iops值高達 256,000。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的區io2塊快速磁碟區 (預覽中)。

volume_size

(選擇性) 以 GiB 為單位指定要建立的磁碟區大小 (如果您不使用快照)。

預設值和支援的值會依據不同而有所不同volume_type。

volume_type = standard

預設值 volume_size = 20 GiB

支援的值 volume_size = GiB

volume_type= gp2 io1、io2、和 gp3

預設值 volume_size = 20 GiB

支援的值 volume_size =

volume_type= sc1 和 st1

預設值 volume_size = 500 GiB

支援的值 volume_size =

```
volume_size = 20
```

> **Note**
>
> 在 AWS ParallelCluster 版本 2.10.1 之前，所有磁碟區類型的預設值都是 20 GiB。

更新政策：如果變更此設定，則不允許更新。

volume_throughput

(選擇性) 定義gp3磁碟區類型的輸送量，以 MIB/s 為單位。

預設值為 125。

支援的值 volume_throughput = 125 至 1000 兆位元/秒

與的比率volume_iops可volume_throughput以不超過 0.25。1000 Mb/s 的最大輸送量需要volume_iops設定至少為 4000。

```
volume_throughput = 1000
```

> **Note**
>
> 在 AWS ParallelCluster 版本 2.10.1 中添加`volume_throughput`了對的 Support。

更新政策：如果變更此設定，則不允許更新。

`volume_type`

(選擇性) 指定您要啟動之磁碟區的 Amazon EBS 磁碟區類型。

有效的選項包括以下磁碟區類型：

`gp2`, `gp3`

通用固態硬碟

`io1`, `io2`

Provisioned IOPS SSD

`st1`

輸送量最佳化 HDD

`sc1`

Cold HDD

`standard`

前一代磁性

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EBS 磁碟區類型。

預設值為 `gp2`。

```
volume_type = io2
```

> **Note**
>
> 在 2.10.1 AWS ParallelCluster 版本中添加`io2`了 Support `gp3` 並添加了。

更新政策：如果變更此設定，則不允許更新。

[efs] 區段

定義掛接在主節點和運算節點上之 Amazon EFS 的組態設定。如需詳細資訊，請參閱CreateFileSystem亞馬遜 EFS API 參考中的。

若要了解如何在叢集定義中包含 Amazon EFS 檔案系統，請參閱[cluster] ##/efs_settings。

若要將現有的 Amazon EFS 檔案系統用於獨立於叢集生命週期的長期永久儲存，請指定efs_fs_id。

如果未指定efs_fs_id，請在AWS ParallelCluster建立叢集時從[efs]設定建立 Amazon EFS 檔案系統，並在刪除叢集時刪除檔案系統和資料。

如需詳細資訊，請參閱最佳做法：將叢集移至新叢集 AWS ParallelCluster 次要或修補程式版本。

格式為[efs *efs-name*]。 *efs-name* 必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[efs customfs]
shared_dir = efs
encrypted = false
performance_mode = generalPurpose
```

主題

- efs_fs_id
- efs_kms_key_id
- encrypted
- performance_mode
- provisioned_throughput
- shared_dir
- throughput_mode

efs_fs_id

(選擇性) 定義現有檔案系統的 Amazon EFS 檔案系統 ID。

指定此選項會使除以外的shared_dir所有其他 Amazon EFS 選項失效。

如果您設定此選項，它只支援下列類型的檔案系統：

- 堆疊的可用區域中沒有掛載目標的檔案系統。
- 在堆疊的可用區域中具有現有掛載目標且允許輸入和輸出 NFS 流量的檔案系統0.0.0.0/0。

驗證 efs_fs_id 的例行性檢查，需要 IAM 角色才能擁有以下許可：

- elasticfilesystem:DescribeMountTargets
- elasticfilesystem:DescribeMountTargetSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeNetworkInterfaceAttribute

若要避免錯誤，您必須將這些許可新增至 IAM 角色，或設定 sanity_check = false。

> **Important**
>
> 當您設定允許輸入和輸出 NFS 流量的掛載目標時0.0.0.0/0，它會將檔案系統公開給掛載目標可用區域中任何位置的 NFS 掛載要求。 AWS不建議在堆疊的可用區域中建立掛載目標。相反，讓我們AWS處理這一步。如果您想要在堆疊的可用區域中有掛載目標，請考慮使用自訂安全性群組，方法是在區[vpc]段下方提供vpc_security_group_id選項。然後，將該安全性群組新增至裝載目標，並關閉sanity_check以建立叢集。

沒有預設值。

```
efs_fs_id = fs-12345
```

更新政策：如果變更此設定，則不允許更新。

efs_kms_key_id

(選擇性) 識別要用來保護加密檔案系統的 AWS Key Management Service (AWS KMS) 客戶管理金鑰。如果設定此選項，則必須將 encrypted 設定設為 true。這對應於亞馬遜 EFS API KmsKeyId參考中的參數。

沒有預設值。

```
efs_kms_key_id = 1234abcd-12ab-34cd-56ef-1234567890ab
```

更新政策：如果變更此設定，則不允許更新。

encrypted

(選擇性) 指出檔案系統是否已加密。這對應於 Amazon EFS API 參考資料中的加密參數。

預設值為 false。

```
encrypted = true
```

更新政策：如果變更此設定，則不允許更新。

performance_mode

(選擇性) 定義檔案系統的效能模式。這對應於亞馬遜 EFS API PerformanceMode參考中的參數。

有效的選項為下列值：

- generalPurpose
- maxIO

這兩個值都區分大小寫。

我們建議對大部分檔案系統使用 generalPurpose 效能模式。

使用 maxIO 效能模式的檔案系統可擴展到更高階的彙總輸出量和每秒操作數。但是，對於大多數文件操作來說，延遲時間會稍高一些。

建立檔案系統之後，就無法變更此參數。

預設值為 generalPurpose。

```
performance_mode = generalPurpose
```

更新政策：如果變更此設定，則不允許更新。

provisioned_throughput

(選擇性) 定義檔案系統的佈建輸送量，以 Mb/s 為單位。這對應於亞馬遜 EFS API ProvisionedThroughputInMibps參考中的參數。

如果使用此參數，則必須將 throughput_mode 設為 provisioned。

輸送量的配額為 1024 Mb/s。若要要求增加配額，請聯絡 AWS Support。

最小值為 0.0 MiB/s。

```
provisioned_throughput = 1024
```

更新政策：此設定可以在更新期間變更。

shared_dir

(必要) 定義主節點和運算節點上的 Amazon EFS 掛載點。

此為必要參數。只有在已指定的情況下，才會使用 Amazon EFS 區段。shared_dir

請勿使用NONE或/NONE作為共用目錄。

下列範例會將亞馬遜 EFS 裝載於/efs。

```
shared_dir = efs
```

更新政策：如果變更此設定，則不允許更新。

throughput_mode

(選擇性) 定義檔案系統的輸送量模式。這對應於亞馬遜 EFS API ThroughputMode參考中的參數。

有效的選項為下列值：

- bursting
- provisioned

預設值為 bursting。

```
throughput_mode = provisioned
```

更新政策：此設定可以在更新期間變更。

[fsx] 區段

定義 Lustre 檔案系統之附加 FSx 的組態設定。如需詳細資訊，請參閱亞馬遜 FSx API 參考資料CreateFileSystem中的亞馬遜 FSx。

如果支援光澤的base_os是alinux2ubuntu2004、、或、FSx。centos7 ubuntu1804

使用亞馬遜 Linux 時，內核必須是4.14.104-78.84.amzn1.x86_64或更新版本。如需指示，請參閱亞馬遜 FSx 用戶指南中的安裝光澤用戶端。

> **Note**
>
> 當awsbatch作排程器使用時，目前不支援 FSx for Lustre。

> **Note**
>
> 在版本 2.10.4 中AWS ParallelCluster移除centos8了對於光澤的 FSx 支援。在AWS ParallelCluster版本 2.11.0 中新增了對於光澤ubuntu2004的 FSx 支援。在AWS ParallelCluster版本 2.10.0 中新增了對於光澤centos8的 FSx 支援。對於光澤的 FSx 的支持alinux2，ubuntu1604並在版本 2.6.0 中AWS ParallelCluster添加ubuntu1804了。在AWS ParallelCluster版本 2.4.0 中增加了對光澤的 centos7 FSx 支持。

如果使用現有的檔案系統，則它必須與安全群組建立關聯，允許傳入 TCP 流量經過連接埠 988。在安全群組規則0.0.0.0/0上將來源設定為，可針對該規則的通訊協定和連接埠範圍，從 VPC 安全性群組內的所有 IP 範圍提供用戶端存取。若要進一步限制對檔案系統的存取，建議您針對安全性群組規則使用更嚴格的來源。例如，您可以使用更特定的 CIDR 範圍、IP 位址或安全性群組識別碼。這不會使用 vpc_security_group_id 而自動完成。

若要將現有的 Amazon FSx 檔案系統用於獨立於叢集生命週期的長期永久儲存，請指定fsx_fs_id。

如果未指定fsx_fs_id，則會在AWS ParallelCluster建立叢集時從 Lustre 檔案系統的[fsx]設定建立 FSx，並在刪除叢集時刪除檔案系統和資料。

如需詳細資訊，請參閱最佳做法：將叢集移至新叢集 AWS ParallelCluster 次要或修補程式版本。

格式為[fsx *fsx-name*]。 *fsx-name* 必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[fsx fs]
shared_dir = /fsx
fsx_fs_id = fs-073c3803dca3e28a6
```

若要建立和設定新的檔案系統，請使用下列參數：

```
[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
import_path = s3://bucket
weekly_maintenance_start_time = 1:00:00
```

主題

- auto_import_policy
- automatic_backup_retention_days
- copy_tags_to_backups
- daily_automatic_backup_start_time
- data_compression_type
- deployment_type
- drive_cache_type
- export_path
- fsx_backup_id
- fsx_fs_id
- fsx_kms_key_id
- import_path
- imported_file_chunk_size
- per_unit_storage_throughput
- shared_dir
- storage_capacity

- storage_type
- weekly_maintenance_start_time

auto_import_policy

(選擇性) 指定自動匯入政策，以反映 S3 儲存貯體中用來建立 Lustre 檔案系統 FSx 的變更。可能值如下：

NEW

FSx for Lustre 會自動匯入新增至連結 S3 儲存貯體 (目前不存在於 Lustre 檔案系統的 FSx 中) 的任何新物件的目錄清單。

NEW_CHANGED

FSx for Lustre 會自動匯入新增至 S3 儲存貯體的任何新物件的檔案和目錄清單，以及 S3 儲存貯體中已變更的任何現有物件。

這對應於AutoImportPolicy屬性。如需詳細資訊，請參閱 Amazon FSx for Lustre 使用者指南中的自動從 S3 儲存貯體匯入更新。指定auto_import_policy參數時，不能指定copy_tags_to_backupsdaily_automatic_backup_start_time、、和fsx_backup_id參數。automatic_backup_retention_days

如果未指定auto_import_policy設定，則會停用自動匯入。FSx for Lustre 只會在建立檔案系統時，從連結的 S3 儲存貯體更新檔案和目錄清單。

```
auto_import_policy = NEW_CHANGED
```

> **Note**
>
> 在AWS ParallelCluster版本 2.10.0 中添加auto_import_policy了對的支持。

更新政策：如果變更此設定，則不允許更新。

automatic_backup_retention_days

(選擇性) 指定保留自動備份的天數。這僅適用於PERSISTENT_1部署類型。指定automatic_backup_retention_days參數時，不能指定export_pathimport_path、、

和imported_file_chunk_size參數。auto_import_policy這對應於AutomaticBackupRetentionDays屬性。

預設值為 0。此設定會停用自動備份。可能的值是介於 0 到 35 (含) 之間的整數。

```
automatic_backup_retention_days = 35
```

> **Note**
>
> 在AWS ParallelCluster版本 2.8.0 中添加automatic_backup_retention_days了對的支持。

更新政策：此設定可以在更新期間變更。

copy_tags_to_backups

(選擇性) 指定是否將檔案系統的標籤複製到備份。這僅適用於PERSISTENT_1部署類型。指定copy_tags_to_backups參數時，automatic_backup_retention_days必須使用大於 0 的值來指定，且不能指定import_path、、和imported_file_chunk_size參數。auto_import_policy export_path這對應於CopyTagsToBackups屬性。

預設值為 false。

```
copy_tags_to_backups = true
```

> **Note**
>
> 在AWS ParallelCluster版本 2.8.0 中添加copy_tags_to_backups了對的支持。

更新政策：如果變更此設定，則不允許更新。

daily_automatic_backup_start_time

(選擇性) 指定開始自動備份的時間 (UTC)。這僅適用於PERSISTENT_1部署類型。指定daily_automatic_backup_start_time參數時，automatic_backup_retention_days必

須使用大於 0 的值來指定，且不能指定import_path、、和imported_file_chunk_size參數。auto_import_policy export_path這對應於DailyAutomaticBackupStartTime屬性。

格式為HH:MM，其中HH是一天中的零填充小時（0-23），並且MM是小時中的零填充分鐘。例如，世界標準時間上午 1:03 如下。

```
daily_automatic_backup_start_time = 01:03
```

預設值是介於和之間的隨機時00:00間23:59。

> **Note**
>
> 在AWS ParallelCluster版本 2.8.0 中添加daily_automatic_backup_start_time了對的支持。

更新政策：此設定可以在更新期間變更。

data_compression_type

(選擇性) 指定 FSx 作為 Lustre 資料壓縮類型。這對應於DataCompressionType屬性。如需詳細資訊，請參閱 Amazon FSx 適用於 Lustre 的資料壓縮使用者指南。

唯一有效的值為 LZ4。若要停用資料壓縮，請移除data_compression_type參數。

```
data_compression_type = LZ4
```

> **Note**
>
> 在AWS ParallelCluster版本 2.11.0 中增加data_compression_type了對的支持。

更新政策：此設定可以在更新期間變更。

deployment_type

(選擇性) 為 Lustre 部署類型指定 FSx。這對應於DeploymentType屬性。如需詳細資訊，請參閱《亞馬遜 FSx 的光澤使用者指南》中的 FSx 部署選項。選擇暫存部署類型，用於暫時儲存和資料的短期處

理。 SCRATCH_2是最新一代的暫存檔案系統。與基準輸送量和資料傳輸中加密相比，它提供了更高的突發輸送量。

有效值為 SCRATCH_1、SCRATCH_2 和 PERSISTENT_1。

SCRATCH_1

Lustre 的 FSx 預設部署類型。使用此部署類型，storage_capacity 設定值可能為 1200、2400 和任何 3600 的倍數。在AWS ParallelCluster版本 2.4.0 中添加了對的SCRATCH_1支持。

SCRATCH_2

最新一代的暫存檔案系統。對於尖峰工作負載，它最多支援六倍的基準輸送量。它也支援支援中支援執行個體類型的資料傳輸加密。AWS 區域如需詳細資訊，請參閱 Amazon FSx for Lustre 使用者指南中的加密傳輸中的資料。使用此部署類型，storage_capacity 設定值可能為 1200 和任何 2400 的倍數。在 2.6.0 AWS ParallelCluster 版本中添加了支持。SCRATCH_2

PERSISTENT_1

專為長期儲存而設計。檔案伺服器具有高可用性，而且資料會在檔案系統的可AWS用區域內複製。它支援支援執行個體類型的傳輸中資料加密。使用此部署類型，storage_capacity 設定值可能為 1200 和任何 2400 的倍數。在 2.6.0 AWS ParallelCluster 版本中添加了支持。PERSISTENT_1

預設值為 SCRATCH_1。

```
deployment_type = SCRATCH_2
```

> **Note**
>
> 在 2.6.0 AWS ParallelCluster 版本中添加了支持。deployment_type

更新政策：如果變更此設定，則不允許更新。

drive_cache_type

(選擇性) 指定檔案系統具有 SSD 磁碟機快取。只有在設定設定為時，才能storage_type設定此選項HDD。這對應於DriveCacheType屬性。如需詳細資訊，請參閱《亞馬遜 FSx 的光澤使用者指南》中的 FSx 部署選項。

唯一有效的值為 READ。若要停用 SSD 磁碟機快取，請勿指定drive_cache_type設定。

```
drive_cache_type = READ
```

> **Note**
>
> 在AWS ParallelCluster版本 2.10.0 中添加drive_cache_type了對的支持。

更新政策：如果變更此設定，則不允許更新。

export_path

(選擇性) 指定匯出檔案系統根目錄的 Amazon S3 路徑。指定export_path參數時，不能指定copy_tags_to_backupsdaily_automatic_backup_start_time、、和fsx_backup_id參數。automatic_backup_retention_days這對應於ExportPath屬性。檔案資料和中繼資料不會自動匯出至export_path. 如需匯出資料和中繼資料的相關資訊，請參閱 Amazon FSx for Lustre 使用者指南中的將變更匯出至資料儲存庫。

預設值是 s3://*import-bucket*/FSxLustre*[creation-timestamp]*，其中 *import-bucket* 為 import_path 參數中提供的儲存貯體。

```
export_path = s3://bucket/folder
```

更新政策：如果變更此設定，則不允許更新。

fsx_backup_id

(選擇性) 指定用於從現有備份還原檔案系統的備份 ID。指定fsx_backup_id參數時auto_import_policydeployment_type，不能指定export_pathfsx_kms_key_idimport_pathimported_file_chunk_size、storage_capacity、和per_unit_storage_throughput參數。這些參數是從備份中讀取的。此外 auto_import_policyexport_path，不能指定import_path、、和imported_file_chunk_size參數。

這對應於BackupId屬性。

```
fsx_backup_id = backup-fedcba98
```

> **Note**
>
> 在AWS ParallelCluster版本 2.8.0 中添加fsx_backup_id了對的支持。

更新政策：如果變更此設定，則不允許更新。

fsx_fs_id

(選擇性) 為 Lustre 檔案系統附加現有的 FSx。

如果指定此選項，則只會使用[fsx]區段中的shared_dir和fsx_fs_id設定，並忽略[fsx]區段中的任何其他設定。

```
fsx_fs_id = fs-073c3803dca3e28a6
```

更新政策：如果變更此設定，則不允許更新。

fsx_kms_key_id

(選擇性) 指定您 AWS Key Management Service (AWS KMS) 客戶受管金鑰的金鑰 ID。

系統會使用此金鑰來加密靜態檔案系統中的資料。

此 ID 必須與自訂 ec2_iam_role 搭配使用。如需詳細資訊，請參閱使用自訂 KMS 金鑰進行磁碟加密。這對應於亞馬遜 FSx API KmsKeyId參考中的參數。

```
fsx_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

> **Note**
>
> 在 2.6.0 AWS ParallelCluster 版本中添加了支持。fsx_kms_key_id

更新政策：如果變更此設定，則不允許更新。

import_path

(選擇性) 指定要從檔案系統載入資料並做為匯出儲存貯體的 S3 儲存貯體。如需詳細資訊，請參閱export_path。如果指定import_path參數

automatic_backup_retention_dayscopy_tags_to_backups，則不能指定daily_automatic_backup_start_time、、和fsx_backup_id參數。這對應於亞馬遜 FSx API ImportPath參考中的參數。

在建立叢集時進行匯入。如需詳細資訊，請參閱 Amazon FSx for Lustre 使用者指南中的從資料儲存庫匯入資料。匯入時，只會匯入檔案中繼資料 (名稱、擁有權、時間戳記和權限)。在首次存取檔案之前，不會從 S3 儲存貯體匯入檔案資料。如需預先載入檔案內容的相關資訊，請參閱 Amazon FSx for Lustre 使用者指南中的將檔案預先載入檔案系統。

如果未提供值，則檔案系統是空的。

```
import_path =  s3://bucket
```

更新政策：如果變更此設定，則不允許更新。

imported_file_chunk_size

(選擇性) 針對從資料儲存庫匯入的檔案 (使用)，決定儲存在單一實體磁碟上的每個檔案 (以 MiB 為單位import_path) 的資料分割計數和最大資料量。單一檔案可分割範圍的磁碟數上限會受組成檔案系統的磁碟總數所限。指定imported_file_chunk_size參數時，不能指定copy_tags_to_backupsdaily_automatic_backup_start_time、、和fsx_backup_id參數。automatic_backup_retention_days這對應於ImportedFileChunkSize屬性。

區塊大小預設值為 1024 (1 GiB)，而且最高可達 512,000 MiB (500 GiB)。Amazon S3 物件大小的上限為 5 TB。

```
imported_file_chunk_size = 1024
```

更新政策：如果變更此設定，則不允許更新。

per_unit_storage_throughput

(為 **PERSISTENT_1** 部署類型的必要項目) 對於 deployment_type = PERSISTENT_1 部署類型，描述 MB/s/TiB 中每個 1 個 TB 位元組 (TiB) 儲存體的讀取和寫入輸送量。檔案系統輸送量的計算方式是將網路系統儲存容量 (TiB) 乘以 per_unit_storage_throughput (MB/s/TiB)。對於 2.4 TiB 的檔案系統，佈建 50 MB/s/TiB 的 per_unit_storage_throughput 可產生 120 MB/s 的檔案系統輸送量。您需要支付佈建輸送量的費用。這對應於PerUnitStorageThroughput屬性。

可能的值取決於storage_type設定的值。

storage_type = SSD

可能值為 50、100、200。

storage_type = HDD

可能的值為 12、40。

```
per_unit_storage_throughput = 200
```

> **Note**
>
> 在 2.6.0 AWS ParallelCluster 版本中添加了支持。per_unit_storage_throughput

更新政策：如果變更此設定，則不允許更新。

shared_dir

(必要) 在頭部和計算節點上定義 Lustre 檔案系統 FSx 的掛載點。

請勿使用NONE或/NONE作為共用目錄。

以下範例會在 /fsx 掛載檔案系統。

```
shared_dir = /fsx
```

更新政策：如果變更此設定，則不允許更新。

storage_capacity

(必要) 指定檔案系統的儲存容量 (GiB)。這對應於StorageCapacity屬性。

儲存容量可能值會依 deployment_type 設定而有所不同。

SCRATCH_1

可能的值是 1200、2400 和任何 3600 的倍數。

SCRATCH_2

可能的值是 1200 和任何 2400 的倍數。

PERSISTENT_1

可能的值會根據其他設定的值而有所不同。

storage_type = SSD

可能的值是 1200 和任何 2400 的倍數。

storage_type = HDD

可能的值會根據設定的設定而有所per_unit_storage_throughput不同。

per_unit_storage_throughput = 12

可能的值是 6000 的任意倍數。

per_unit_storage_throughput = 40

可能的值是 1800 的任意倍數。

```
storage_capacity = 7200
```

> **Note**
>
> 對於AWS ParallelCluster版本 2.5.0 和 2.5.1，storage_capacity支援的可能值為 1200、2400，以及任何 3600 的倍數。對於版本 2.5.0 之前的AWS ParallelCluster版本storage_capacity，最小大小為 3600。

更新政策：如果變更此設定，則不允許更新。

storage_type

(選擇性) 指定檔案系統的儲存類型。這對應於StorageType屬性。可能的值為 SSD 和 HDD。預設值為 SSD。

儲存區類型會變更其他設定的可能值。

storage_type = SSD

指定已售狀態磁碟機 (SSD) 儲存區類型。

storage_type = SSD變更數個其他設定的可能值。

drive_cache_type

無法指定此設定。

deployment_type

此設定可設定為SCRATCH_1SCRATCH_2、或PERSISTENT_1。

per_unit_storage_throughput

如果 deployment_type 設定為 PERSISTENT_1，則必須指定此設定。可能的值為 50、100 或 200。

storage_capacity

必須指定此設定。可能的值會根據不同而有所不同deployment_type。

deployment_type = SCRATCH_1

storage_capacity可以是 1200、2400 或任何 3600 的倍數。

deployment_type = SCRATCH_2 或 deployment_type = PERSISTENT_1

storage_capacity可以是 1200 或 2400 的任何倍數。

storage_type = HDD

指定硬碟 (HDD) 儲存類型。

storage_type = HDD變更其他設定的可能值。

drive_cache_type

可以指定此設定。

deployment_type

此設定必須設定為PERSISTENT_1。

per_unit_storage_throughput

必須指定此設定。可能的值為 12 或 40。

storage_capacity

必須指定此設定。可能的值會根據per_unit_storage_throughput設定而有所不同。

```
storage_capacity = 12
```

storage_capacity可以是 6000 的任意倍數。

```
storage_capacity = 40
```

storage_capacity可以是 1800 的任意倍數。

```
storage_type = SSD
```

> **Note**
>
> 在AWS ParallelCluster版本 2.10.0 中添加了對該storage_type設置的支持。

更新政策：如果變更此設定，則不允許更新。

weekly_maintenance_start_time

(選用) 以 UTC 時區指定偏好的每週維護執行時間。這對應於WeeklyMaintenanceStartTime屬性。

格式為 [星期幾]:[小時]:[分鐘]。例如，星期一午夜如下。

```
weekly_maintenance_start_time = 1:00:00
```

更新政策：此設定可以在更新期間變更。

[queue] 區段

定義單一佇列的組態設定。 [queue]僅當設定為時scheduler，才支援剖面slurm。

格式為[queue *<queue-name>*]。 *####*必須以小寫字母開頭，不得超過 30 個字元，且只能包含小寫字母、數字和連字號 (-)。

```
[queue q1]
compute_resource_settings = i1,i2
placement_group = DYNAMIC
enable_efa = true
disable_hyperthreading = false
```

```
compute_type = spot
```

> **Note**
>
> 在 AWS ParallelCluster 版本 2.9.0 中添加了對該[queue]部分的 Support。

主題

compute_resource_settings

(必要) 識[compute_resource]別包含此佇列之計算資源組態的區段。段落名稱必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

每個區[compute_resource]段最多支援三 (3) 個[queue]區段。

例如，下列設定指定開始[compute_resource cr1]和使用[compute_resource cr2]的區段。

```
compute_resource_settings = cr1, cr2
```

更新政策：如果變更此設定，則不允許更新。

compute_type

(選擇性) 定義要針對此佇列啟動的執行個體類型。此設定會取代 cluster_type 設定。

有效選項為：ondemand 和 spot。

預設值為 ondemand。

如需 Spot 執行個體的詳細資訊，請參閱 使用 競價型執行個體。

> **Note**
>
> 使用 Spot 執行個體時，您的帳戶中必須有`AWSServiceRoleForEC2Spot`服務連結角色。若要使用在您的帳戶中建立此角色 AWS CLI，請執行下列命令：
>
> ```
> aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
> ```
>
> 如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Spot 執行個體請求的服務連結角色。

下列範例會用 SpotInstances 於此佇列中的運算節點。

```
compute_type = spot
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

disable_hyperthreading

(選擇性) 停用此佇列中節點上的超執行緒。並非所有執行個體類型都可以停用超執行緒。如需支援停用超執行緒的執行個體類型清單，請參閱 Amazon EC2 使用者指南中每個執行個體類型每個 CPU 核心的 CPU 核心和執行緒。如果已定`disable_hyperthreading`義`[cluster]`區段中的設定，則無法定義此設定。

預設值為 `false`。

```
disable_hyperthreading = true
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

enable_efa

(選擇性) 如果設定為`true`，則指定針對此佇列中的節點啟用 Elastic Fabric Adapter (EFA)。若要檢視支援 EFA 的 EC2 執行個體清單，請參閱 Amazon EC2 Linux 執行個體使用者指南中支援的執行個體類型。如果已定`enable_efa`義`[cluster]`區段中的設定，則無法定義此設定。應使用叢集置放群組以充分減少執行個體之間的延遲。如需詳細資訊，請參閱 `placement` 及 `placement_group`。

```
enable_efa = true
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

enable_efa_gdr

(選擇性) 從 2.11.3 AWS ParallelCluster 版開始，此設定沒有任何作用。如果執行個體類型支援運算節點，則會一律啟用 GPUDirect RDMA (遠端直接記憶體存取) 的彈性網狀架構介面卡 (EFA) 支援。

> **Note**
>
> AWS ParallelCluster 2.10.0 到 2.11.2 版：如果`true`，指定為此佇列中的節點啟用 Elastic Fabric Adapter (EFA) GPUDirect RDMA (遠端直接記憶體存取)。若`true`要將此設定`enable_efa`設定為 `true` .EFA GPUDirect RDMA，這些作業系統上的下列執行個體類型 (`p4d.24xlarge`) 支援這些作業系統 (`alinux2`、、或)。`centos7` `ubuntu1804` `ubuntu2004`如果已定`enable_efa_gdr`義`[cluster]`區段中的設定，則無法定義此設定。應使用叢集置放群組以充分減少執行個體之間的延遲。如需詳細資訊，請參閱 `placement` 及 `placement_group`。

預設值為 `false`。

```
enable_efa_gdr = true
```

> **Note**
>
> 在 AWS ParallelCluster 版本 2.10.0 中添加`enable_efa_gdr`了對的 Support。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

placement_group

(選擇性) 如果存在，則定義此佇列的放置群組。此設定會取代 `placement_group` 設定。

有效的選項為下列值：

- `DYNAMIC`
- 現有的 Amazon EC2 叢集置放群組名稱

設定為時`DYNAMIC`，會建立此佇列的唯一放置群組，並做為叢集堆疊的一部分刪除。

如需置放群組的詳細資訊，請參閱 Amazon EC2 使用者指南中的放置群組。如果不同的執行個體類型使用相同的置放群組，則要求可能會因為容量不足錯誤而失敗。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的執行個體容量不足。

沒有預設值。

並非所有執行個體類型都支援叢集置放群組。例如，t2.micro不支援叢集置放群組。如需支援叢集置放群組的執行個體類型清單的相關資訊，請參閱 Amazon EC2 使用者指南中的叢集放置群組規則和限制。如需使用置放群組的秘訣，請參閱 置放群組和執行個體啟動問題。

```
placement_group = DYNAMIC
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

[raid] 區段

定義由多個相同 Amazon EBS 磁碟區建立的 RAID 陣列的組態設定。RAID 磁碟機掛接在頭節點上，並匯出至含 NFS 的計算節點。

格式為[raid *raid-name*]。*raid-name* 必須以字母開頭，不得超過 30 個字元，且只能包含字母、數字、連字號 (-) 和底線 (_)。

```
[raid rs]
shared_dir = raid
raid_type = 1
num_of_raid_volumes = 2
encrypted = true
```

主題

- shared_dir
- ebs_kms_key_id
- encrypted
- num_of_raid_volumes
- raid_type
- volume_iops
- volume_size
- volume_throughput

- volume_type

shared_dir

(必要) 定義磁頭和運算節點上 RAID 陣列的掛載點。

只在指定此參數時，才會建立 RAID 磁碟機。

請勿使用NONE或/NONE作為共用目錄。

以下範例會在 /raid 掛載陣列。

```
shared_dir = raid
```

更新政策：如果變更此設定，則不允許更新。

ebs_kms_key_id

(選擇性) 指定用於加密的自訂 AWS KMS 金鑰。

此參數必須與 encrypted = true 搭配使用，而且它必須具有自訂 ec2_iam_role。

如需詳細資訊，請參閱 使用自訂 KMS 金鑰進行磁碟加密。

```
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

更新政策：如果變更此設定，則不允許更新。

encrypted

(選擇性) 指定檔案系統是否已加密。

預設值為 false。

```
encrypted = false
```

更新政策：如果變更此設定，則不允許更新。

num_of_raid_volumes

(選擇性) 定義要從中組合 RAID 陣列的 Amazon EBS 磁碟區數量。

磁碟區的最小數目為2。

磁碟區的最大數目為5。

預設值為 2。

```
num_of_raid_volumes = 2
```

更新政策：如果變更此設定，則不允許更新。

raid_type

(必要) 定義 RAID 陣列的 RAID 類型。

只在指定此參數時，才會建立 RAID 磁碟機。

有效選項為下列值：

- 0
- 1

如需 RAID 類型的詳細資訊，請參閱 Amazon EC2 使用者指南中的 RAID 資訊。

以下範例會建立 RAID 0 陣列：

```
raid_type = 0
```

更新政策：如果變更此設定，則不允許更新。

volume_iops

(選擇性) 定義io1、io2和gp3類型磁碟區的 IOPS 數目。

預設值、支援的值和總成volume_size比例會volume_iops隨和而有所volume_type不同volume_size。

volume_type = io1

預設值 volume_iops = 100

支援的值 volume_iops = 100 至 64 萬 †

最大volume_iops比volume_size率 = 每 GiB 50 IOPS。5000 IOPS 需要一volume_size個至少 100 GiB。

volume_type = io2

預設值 volume_iops = 100

支援的值 volume_iops = 100 至 64 萬 (對於io2區塊快速磁碟區為 256000) †

最大volume_iops比volume_size率 = 每 GiB 500 IOPS。5000 IOPS 需要一個volume_size至少 10 GiB 博。

volume_type = gp3

預設值 volume_iops =

支援的volume_iops值：

最大volume_iops比volume_size率 = 每 GiB 500 IOPS。5000 IOPS 需要一個volume_size至少 10 GiB 博。

```
volume_iops = 3000
```

更新政策：此設定可以在更新期間變更。

† 只有在佈建超過 32,000 IOPS 的硝基系統上建置的執行個體上才能保證最大 IOPS。其他執行個體可保證高達 32,000 IOPS。除非您修改io1磁碟區，否則舊磁碟區可能無法達到完整效能。 io2區塊快速磁碟區支援的volume_iops值高達 256000。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的區io2塊快速磁碟區 (預覽中)。

volume_size

(選擇性) 以 GiB 為單位定義要建立的磁碟區大小。

預設值和支援的值會依據不同而有所不同volume_type。

volume_type = standard

預設值 volume_size = 20 GiB

支援的值 volume_size = GiB

volume_type= gp2 io1、io2、和 gp3

預設值 volume_size = 20 GiB

支援的值 volume_size =

volume_type= sc1 和 st1

預設值 volume_size = 500 GiB

支援的值 volume_size =

```
volume_size = 20
```

> **Note**
>
> 在 AWS ParallelCluster 版本 2.10.1 之前，所有磁碟區類型的預設值都是 20 GiB。

更新政策：如果變更此設定，則不允許更新。

volume_throughput

(選擇性) 定義gp3磁碟區類型的輸送量，以 MIB/s 為單位。

預設值為 125。

支援的值 volume_throughput = 125 至 1000 兆位元/秒

與的比率volume_iops可volume_throughput以不超過 0.25。1000 Mb/s 的最大輸送量需要volume_iops設定至少為 4000。

```
volume_throughput = 1000
```

> **Note**
>
> 在 AWS ParallelCluster 版本 2.10.1 中添加volume_throughput了對的 Support。

更新政策：如果變更此設定，則不允許更新。

volume_type

(選擇性) 定義要建置的磁碟區類型。

有效選項為下列值：

gp2, gp3

通用固態硬碟

io1, io2

Provisioned IOPS SSD

st1

輸送量最佳化 HDD

sc1

Cold HDD

standard

前一代磁性

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Amazon EBS 磁碟區類型。

預設值為 gp2。

```
volume_type = io2
```

> **Note**
>
> 在 2.10.1 AWS ParallelCluster 版本中添加io2了 Support gp3 並添加了。

更新政策：如果變更此設定，則不允許更新。

[scaling] 區段

主題

- scaledown_idletime

指定會定義運算節點擴展方式的設定。

格式為[scaling *scaling-name*]。 *####*必須以字母開頭，不得超過 30 個字元，且僅包含字母、數字、連字號 (-) 和底線 (_)。

```
[scaling custom]
scaledown_idletime = 10
```

scaledown_idletime

(選擇性) 指定沒有工作的時間長度 (以分鐘為單位)，計算節點在此之後終止。

如果awsbatch是排程器，則不會使用此參數。

預設值為 10。

```
scaledown_idletime = 10
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

[vpc] 區段

指定 Amazon VPC組態設定。如需 的詳細資訊VPCs，請參閱 Amazon 使用者指南 中的什麼是 AmazonVPC？ 和 安全最佳實務VPC。 VPC

格式是 [vpc *vpc-name*]。*vpc-name* 必須以字母開頭，包含不超過 30 個字元，且僅包含字母、數字、連字號（-）和底線（_）。

```
[vpc public]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-xxxxxx
```

主題

- additional_sg
- compute_subnet_cidr

- compute_subnet_id
- master_subnet_id
- ssh_from
- use_public_ips
- vpc_id
- vpc_security_group_id

additional_sg

（選用）為所有執行個體提供額外的 Amazon VPC安全群組 ID。

沒有預設值。

```
additional_sg = sg-xxxxxx
```

compute_subnet_cidr

（選用）指定無類別網域間路由（CIDR）區塊。如果您想要建立運算子網路 AWS ParallelCluster ，請使用此參數。

```
compute_subnet_cidr = 10.0.100.0/24
```

更新政策：如果變更此設定，則不允許更新。

compute_subnet_id

（選用）指定要在其中佈建運算節點的現有子網路 ID。

如果未指定，則 compute_subnet_id 會使用 master_subnet_id 的值。

如果子網路是私有的，您必須設定 NAT以進行 Web 存取。

```
compute_subnet_id = subnet-xxxxxx
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

master_subnet_id

（必要）指定要在其中佈建主機節點的現有子網路 ID。

```
master_subnet_id = subnet-xxxxxx
```

更新政策：如果變更此設定，則不允許更新。

ssh_from

（選用）指定要允許SSH存取的 CIDR格式化 IP 範圍。

此參數只會在 AWS ParallelCluster 建立安全群組時使用。

預設值為 0.0.0.0/0。

```
ssh_from = 0.0.0.0/0
```

更新政策：此設定可以在更新期間變更。

use_public_ips

（選用）定義是否要將公有 IP 地址指派給運算執行個體。

如果設定為 true，彈性 IP 地址會與主機節點相關聯。

如果設定為 false，則主機節點會根據「自動指派公有 IP」子網路組態參數的值，具有公有 IP（無論是否具有）。

如需範例，請參閱聯網設定。

預設值為 true。

```
use_public_ips = true
```

> **Important**
>
> 依預設，每個 限制 AWS 帳戶 為五（5）個彈性 IP 地址 AWS 區域。如需詳細資訊，請參閱 Amazon EC2 使用者指南 中的彈性 IP 地址限制。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

vpc_id

（必要）指定要在VPC其中佈建叢集的 Amazon ID。

```
vpc_id = vpc-xxxxxx
```

更新政策：如果變更此設定，則不允許更新。

vpc_security_group_id

（選用）指定對所有執行個體使用現有安全群組。

沒有預設值。

```
vpc_security_group_id = sg-xxxxxx
```

由 建立的安全群組 AWS ParallelCluster 允許從ssh_from設定中指定的地址使用連接埠 22 進行SSH存取，如果未指定ssh_from設定，則允許使用所有IPv4地址（0.0.0.0/0）。如果DCV啟用 Amazon，則安全群組允許DCV使用連接埠 8443（或任何port設定指定）從access_from設定中指定的地址存取 Amazon，如果未指定access_from設定，則允許存取所有IPv4地址（0.0.0.0/0）。

> **Warning**
>
> 您可以變更此參數的值，如果 [cluster] fsx_settings 未指定，或同時在 fsx-fs-id中指定 FSx Lustre 檔案系統的外部現有 fsx_settings，則可以更新叢集[fsx fs]。
> 如果在 fsx_settings和 中指定 受 AWS ParallelCluster 管FSx的 Lustre 檔案系統，則無法變更此參數的值[fsx fs]。

更新政策：如果未在組態中指定受 AWS ParallelCluster 管 Amazon FSx for Lustre 檔案系統，則可以在更新期間變更此設定。

範例

下列範例組AWS ParallelCluster態示範使用SlurmTorque、和AWS Batch排程器的組態。

> **Note**
>
> 從版本 2.11.5 開始，AWS ParallelCluster不支持使用SGE或Torque調度程序。

內容

Slurm Workload Manager (**slurm**)

以下範例會啟動具 slurm 排程器的叢集。範例配置會啟動 1 個具有 2 個工作佇列的叢集。第一個佇列最初spot有 2 個可用的 t3.micro Spot 執行個體。它最多可以擴展到 10 個執行個體，並在 10 分鐘內沒有任務執行時縮減至最少 1 個執行個體 (可使用scaledown_idletime設定進行調整)。第二個佇列不會從執行個體開始ondemand，最多可擴充至 5 個t3.micro隨需執行個體。

```
[global]
update_check = true
sanity_check = true
cluster_template = slurm

[aws]
aws_region_name = <your AWS ##>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster slurm]
key_name = <your EC2 keypair name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = t3.micro     # optional, defaults to t3.micro
vpc_settings = public
queue_settings = spot,ondemand

[queue spot]
compute_resource_settings = spot_i1
```

```
compute_type = spot                  # optional, defaults to ondemand

[compute_resource spot_i1]
instance_type = t3.micro
min_count = 1                        # optional, defaults to 0
initial_count = 2                    # optional, defaults to 0

[queue ondemand]
compute_resource_settings = ondemand_i1

[compute_resource ondemand_i1]
instance_type = t3.micro
max_count = 5                        # optional, defaults to 10
```

Son of Grid Engine(**sge**) 和 Torque Resource Manager (**torque**)

> **Note**
>
> 此範例僅適用於AWS ParallelCluster版本 2.11.4 以上且包含在內的版本。從版本 2.11.5 開始，AWS ParallelCluster不支持使用SGE或Torque調度程序。

下列範例會使用torque或sge排程器啟動叢集。若要使用SGE，請變更scheduler = torque為scheduler = sge。此範例組態最多允許 5 個並行節點，並且在 10 分鐘內沒有工作執行時，可縮減為兩個節點。

```
[global]
update_check = true
sanity_check = true
cluster_template = torque

[aws]
aws_region_name = <your AWS ##>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster torque]
key_name = <your EC2 keypair name>but they aren't eligible for future updates
base_os = alinux2                    # optional, defaults to alinux2
scheduler = torque                   # optional, defaults to sge
```

```
master_instance_type = t3.micro     # optional, defaults to t3.micro
vpc_settings = public
initial_queue_size = 2              # optional, defaults to 0
maintain_initial_size = true        # optional, defaults to false
max_queue_size = 5                  # optional, defaults to 10
```

> **Note**
>
> 從版本 2.11.5 開始，AWS ParallelCluster不支持使用SGE或Torque調度程序。如果您使用這些版本，您可以繼續使用它們，或疑難排解AWS服務和AWS支援團隊的支援。

AWS Batch (**awsbatch**)

以下範例會啟動具 awsbatch 排程器的叢集。它設置為根據您的工作資源需求選擇更好的實例類型。

範例組態最多允許 40 個並行 vCPU，並且在 10 分鐘內未執行任務時縮減為零 (可使用此scaledown_idletime設定進行調整)。

```
[global]
update_check = true
sanity_check = true
cluster_template = awsbatch

[aws]
aws_region_name = <your AWS ##>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster awsbatch]
scheduler = awsbatch
compute_instance_type = optimal # optional, defaults to optimal
min_vcpus = 0                   # optional, defaults to 0
desired_vcpus = 0               # optional, defaults to 4
max_vcpus = 40                  # optional, defaults to 20
base_os = alinux2               # optional, defaults to alinux2, controls the base_os
 of
                                # the head node and the docker image for the compute
 fleet
key_name = <your EC2 keypair name>
```

```
vpc_settings = public
```

AWS ParallelCluster 的運作方式

AWS ParallelCluster不僅是為了管理叢集的一種方式而建置，還可做為如何使用AWS服務建置 HPC 環境的參考。

主題

AWS ParallelCluster程序

本節僅適用於以其中一個支援的傳統任務排程器 (SGE、Slurm 或 Torque) 部署的 HPC 叢集。與這些排程器搭配使用時，可透過與 Auto Scaling 群組和基礎工作排程器互動來AWS ParallelCluster管理計算節點佈建和移除。

對於以下項目為基礎的 HPC 叢集AWS Batch，則AWS Batch需AWS ParallelCluster仰賴運算節點管理所提供的功能。

> **Note**
>
> 從版本 2.11.5 開始，AWS ParallelCluster不支持使用SGE或Torque調度程序。您可以在 2.11.4 以下版本中繼續使用它們，但它們不符合AWS服務和支持團隊的未來更新或疑難排解AWS支持的資格。

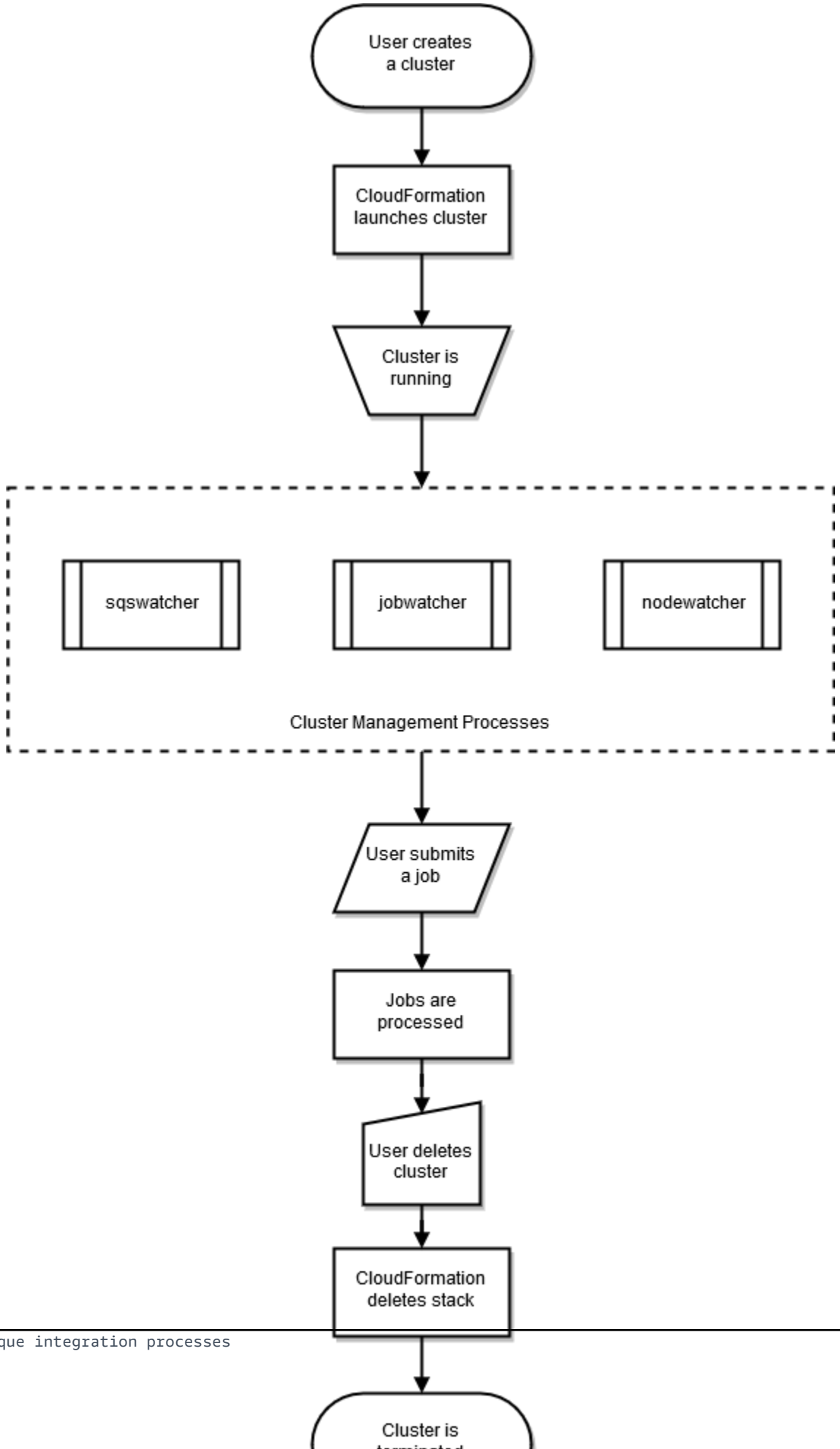
主題

SGE and Torque integration processes

> **Note**
>
> 本節僅適用於包括 2.11.4 版AWS ParallelCluster本的版本。從版本 2.11.5 開始，AWS ParallelCluster不支援使用SGE和Torque排程器、亞馬遜 SNS 和亞馬遜 SQS。

一般概述

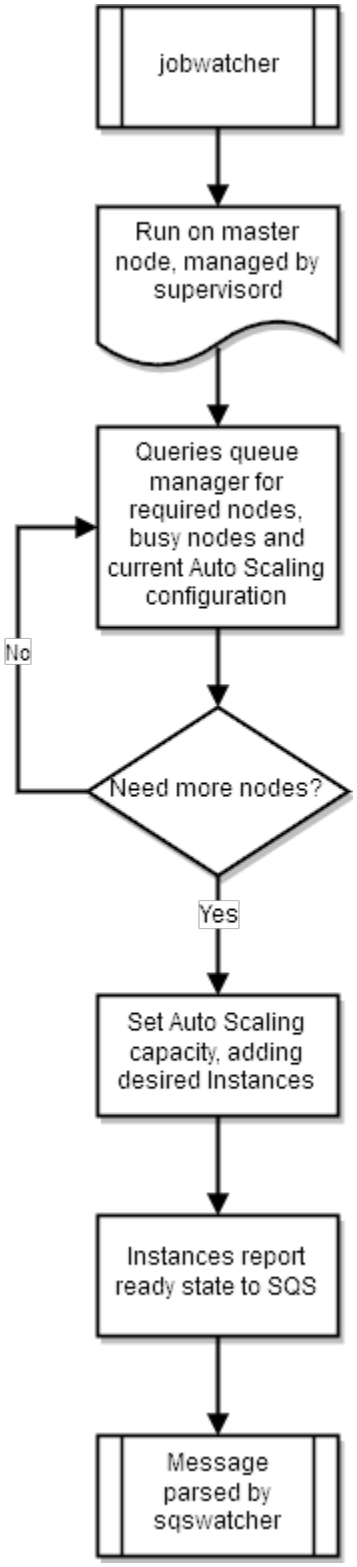
叢集的生命週期在使用者建立叢集之後開始。通常，叢集是從命令列界面 (CLI) 建立的。在建立叢集之後，它會一直存在，直到將其刪除。AWS ParallelCluster 協助程式會在叢集節點上執行，主要管理 HPC 叢集彈性。下圖顯示使用者工作流程和叢集生命週期。以下各節說明用來管理叢集的 AWS ParallelCluster 協助程式。


User creates a cluster
CloudFormation launches cluster
Cluster is running
sqswatcher
jobwatcher
nodewatcher
Cluster Management Processes
User submits a job
Jobs are processed
User deletes cluster
CloudFormation deletes stack
Cluster is

使用SGE和Torque排程器nodewatcherjobwatcher、AWS ParallelCluster使用和sqswatcher程序。

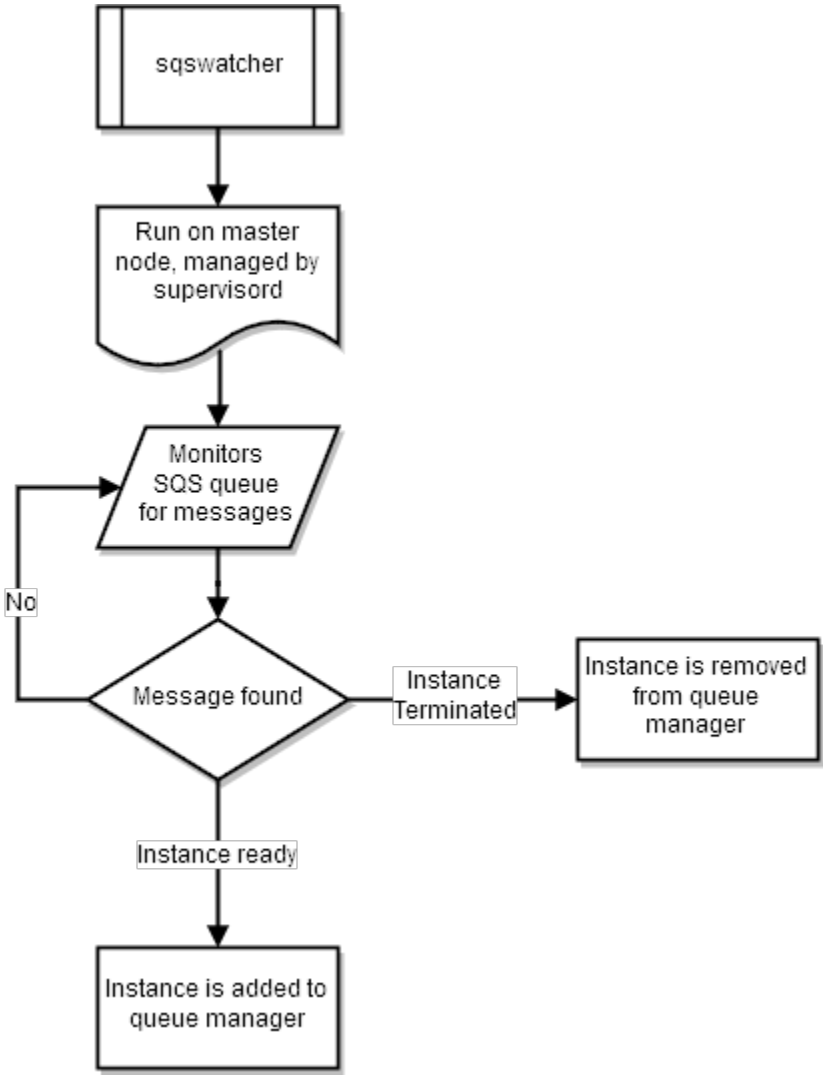
jobwatcher

叢集執行時，root 使用者擁有的處理序會監視已設定的排程器 (SGE或Torque)。它每分鐘評估隊列，以決定何時擴展。


jobwatcher
Run on master node, managed by supervisord
Queries queue manager for required nodes, busy nodes and current Auto Scaling configuration
No
Need more nodes?
Yes
Set Auto Scaling capacity, adding desired Instances
Instances report ready state to SQS
Message parsed by sqswatcher

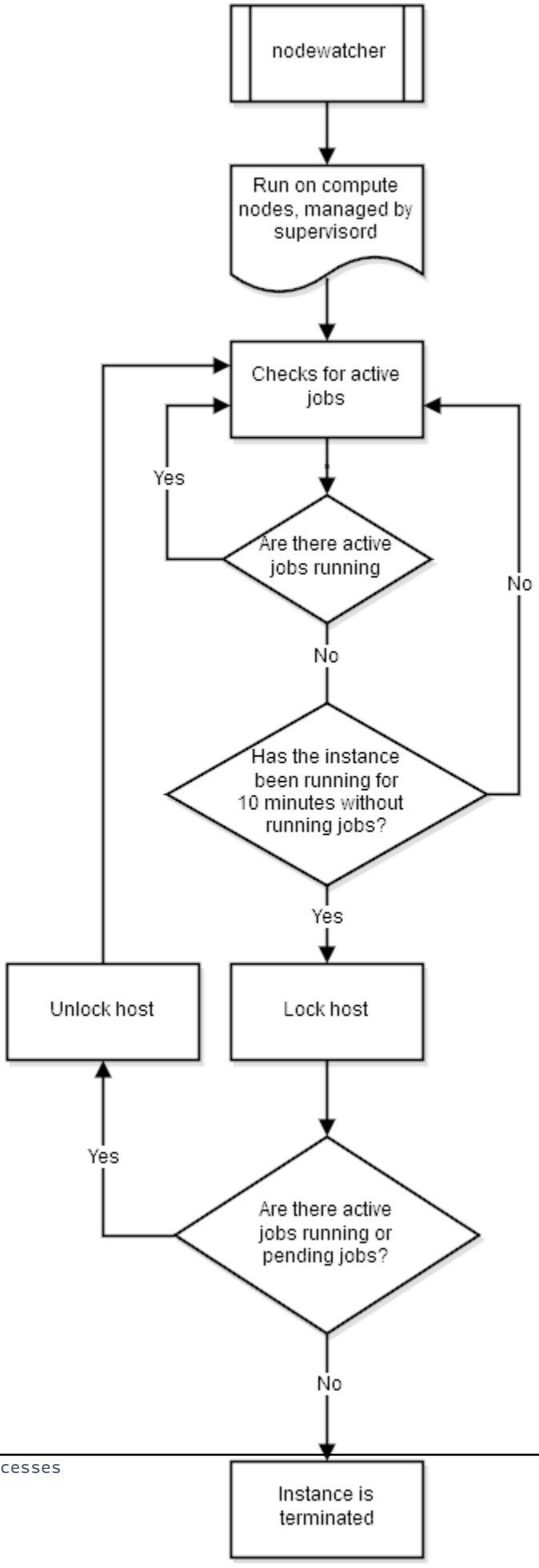
sqswatcher

此sqswatcher程序會監控由 Auto Scaling 傳送的 Amazon SQS 訊息，以通知您叢集內的狀態變更。執行個體上線時，會將「執行個體就緒」訊息提交給 Amazon SQS。在頭節點上執行時sqs_watcher，會接收此訊息。當新執行個體上線或終止時，這些訊息用來通知佇列管理員，因此管理員能夠在佇列中新增或移除執行個體。



nodewatcher

nodewatcher 程序在運算機群中的每個節點上執行。在使用者定義的 scaledown_idletime 期間之後，會終止該執行個體。


nodewatcher
Run on compute nodes, managed by supervisord
Checks for active jobs
Yes
Are there active jobs running
No
No
Has the instance been running for 10 minutes without running jobs?
Yes
Unlock host
Lock host
Yes
Are there active jobs running or pending jobs?
No
Instance is terminated

Slurm integration processes

使用Slurm排程器、AWS ParallelCluster使用clustermgtd和computemgt程序。

clustermgtd

以異質模式執行 (以指定queue_settings值表示) 的叢集具有在標頭節點上執行的叢集管理常駐程式 (clustermgtd) 處理序。這些工作由叢集管理常駐程式執行。

- 非作用中分割區清理
- 靜態容量管理：確保靜態容量始終保持正常運作
- 與亞馬遜 EC2 同步調度程序。
- 清理孤立執行個體
- 在暫停工作流程之外發生的 Amazon EC2 終止上還原排程器節點狀態
- 不健康的亞馬遜 EC2 實例管理（亞馬遜 EC2 運行狀態檢查失敗）
- 排程維護事件管理
- 不健康的排程器節點管理 (排程器健全狀況檢查失敗

computemgtd

以異質模式執行 (以指定queue_settings值表示) 的叢集具有在每個計算節點上執行的計算管理常駐程式 (computemgtd) 處理序。計算管理常駐程式每五 (5) 分鐘就會確認頭節點可以到達且狀態良好。如果經過五 (5) 分鐘，在此期間無法到達頭節點或狀態不佳，則會關閉計算節點。

AWS 所使用的 服務 AWS ParallelCluster

使用下列 Amazon Web Services（AWS）服務 AWS ParallelCluster。

主題

- AWS Auto Scaling
- AWS Batch
- AWS CloudFormation
- Amazon CloudWatch
- Amazon CloudWatch Logs
- AWS CodeBuild

- Amazon DynamoDB
- Amazon Elastic Block Store
- Amazon Elastic Compute Cloud
- Amazon Elastic Container Registry
- Amazon EFS
- Amazon FSx for Lustre
- AWS Identity and Access Management
- AWS Lambda
- Amazon DCV
- Amazon Route 53
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Simple Storage Service
- Amazon VPC

AWS Auto Scaling

> **Note**
>
> 本節僅適用於 2 AWS ParallelCluster .11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 AWS Auto Scaling。

AWS Auto Scaling 是一項服務，可監控您的應用程式，並根據您的特定和不斷變化的服務需求自動調整容量。此服務會以 Auto Scaling 群組的形式管理您的 ComputeFleet 執行個體。群組可以彈性地由不斷變化的工作負載驅動，或由初始執行個體組態靜態修正。

AWS Auto Scaling 會與 ComputeFleet 執行個體搭配使用，但不會與 AWS Batch 叢集搭配使用。

如需 的詳細資訊 AWS Auto Scaling，請參閱 https://aws.amazon.com/autoscaling/和 https://docs.aws.amazon.com/autoscaling/。

AWS Batch

AWS Batch 是 AWS 受管任務排程器服務。它動態佈建 AWS Batch 叢集中運算資源的最佳數量和類型（例如，CPU或記憶體最佳化執行個體）。這些資源會根據批次任務的特定需求佈建，包括磁碟區需求。使用 AWS Batch，您不需要安裝或管理其他批次運算軟體或伺服器叢集，即可有效執行任務。

AWS Batch 僅用於 AWS Batch 叢集。

如需 的詳細資訊 AWS Batch，請參閱 https://aws.amazon.com/batch/和 https://docs.aws.amazon.com/batch/。

AWS CloudFormation

AWS CloudFormation 是一種 infrastructure-as-code 服務，提供通用語言，以在您的雲端環境中建立和佈建 AWS 第三方應用程式資源。這是 使用的主要服務 AWS ParallelCluster。中的每個叢集 AWS ParallelCluster 都會以堆疊表示，且每個叢集所需的所有資源都會在 AWS ParallelCluster AWS CloudFormation 範本中定義。在大多數情況下， AWS ParallelCluster CLI命令會直接對應至 AWS CloudFormation 堆疊命令，例如建立、更新和刪除命令。在叢集中啟動的執行個體會HTTPS呼叫叢集啟動 AWS 區域 所在的 中的 AWS CloudFormation 端點。

如需 的詳細資訊 AWS CloudFormation，請參閱 https://aws.amazon.com/cloudformation/和 https://docs.aws.amazon.com/cloudformation/。

Amazon CloudWatch

Amazon CloudWatch （CloudWatch） 是一種監控和可觀測性服務，可為您提供資料和可操作的洞察。這些洞見可用於監控您的應用程式、回應效能變更和服務例外狀況，以及最佳化資源使用率。在 AWS ParallelCluster中， CloudWatch 用於儀表板，以監控和記錄 Docker 映像建置步驟和 AWS Batch 任務的輸出。

在 2.10.0 AWS ParallelCluster 版之前， CloudWatch 僅與 AWS Batch 叢集搭配使用。

如需 的詳細資訊 CloudWatch，請參閱 https://aws.amazon.com/cloudwatch/和 https://docs.aws.amazon.com/cloudwatch/。

Amazon CloudWatch Logs

Amazon CloudWatch Logs （CloudWatch Logs） 是 Amazon 的核心功能之一 CloudWatch。您可以使用它來監控、儲存、檢視和搜尋 使用的許多元件的日誌檔案 AWS ParallelCluster。

在 2.6.0 AWS ParallelCluster 版之前， CloudWatch Logs 僅與 AWS Batch 叢集搭配使用。

如需詳細資訊，請參閱與 Amazon CloudWatch Logs 整合。

AWS CodeBuild

AWS CodeBuild （ CodeBuild ） 是一種 AWS 受管持續整合服務，符合原始程式碼、執行測試，並產生準備好部署的軟體套件。在 中 AWS ParallelCluster， CodeBuild 用於在建立叢集時自動且透明地建置 Docker 映像。

CodeBuild 僅用於 AWS Batch 叢集。

如需 的詳細資訊 CodeBuild，請參閱 https://aws.amazon.com/codebuild/和 https://docs.aws.amazon.com/codebuild/。

Amazon DynamoDB

Amazon DynamoDB DynamoDB ） 是一種快速且靈活的無SQL資料庫服務。它用於儲存叢集的最小狀態資訊。主節點會追蹤 DynamoDB 資料表中的佈建執行個體。

DynamoDB 不會與 AWS Batch 叢集搭配使用。

如需 DynamoDB 的詳細資訊，請參閱 https://aws.amazon.com/dynamodb/和 https://docs.aws.amazon.com/dynamodb/。

Amazon Elastic Block Store

Amazon Elastic Block Store （ Amazon EBS ） 是一種高效能區塊儲存服務，可為共用磁碟區提供持久性儲存。所有 Amazon EBS設定都可以透過組態傳遞。Amazon EBS磁碟區可以初始化為空白，也可以從現有的 Amazon EBS快照初始化。

如需 Amazon 的詳細資訊EBS，請參閱 https://aws.amazon.com/ebs/和 https://docs.aws.amazon.com/ebs/。

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud （ Amazon EC2 ） 提供 的運算容量 AWS ParallelCluster。主節點和運算節點是 Amazon EC2執行個體。HVM 可以選擇支援的任何執行個體類型。主節點和運算節點可以是不同的執行個體類型。此外，如果使用多個佇列，也可以啟動部分或全部運算節點作為 Spot 執行個體。在執行個體上找到的執行個體存放區磁碟區會掛載為分割磁碟LVM區。

如需 Amazon 的詳細資訊EC2，請參閱 https://aws.amazon.com/ec2/和 https://docs.aws.amazon.com/ec2/。

Amazon Elastic Container Registry

Amazon Elastic Container Registry（Amazon ECR）是完全受管的 Docker 容器登錄檔，可讓您輕鬆儲存、管理和部署 Docker 容器映像。在 中 AWS ParallelCluster，Amazon 會ECR存放建立叢集時建置的 Docker 映像。然後， 會使用 Docker 映像 AWS Batch 來執行已提交任務的容器。

Amazon ECR僅用於 AWS Batch 叢集。

如需詳細資訊，請參閱 https://aws.amazon.com/ecr/和 https://docs.aws.amazon.com/ecr/。

Amazon EFS

Amazon Elastic File System（Amazon EFS）提供簡單、可擴展且完全受管的彈性NFS檔案系統，可用於 AWS 雲端 服務和內部部署資源。指定`efs_settings`設定時，EFS會使用 Amazon，並參考`[efs]`一節。Amazon 的支援EFS已新增至 2.1.0 AWS ParallelCluster 版。

如需 Amazon 的詳細資訊EFS，請參閱 https://aws.amazon.com/efs/和 https://docs.aws.amazon.com/efs/。

Amazon FSx for Lustre

FSx for Lustre 提供使用開放原始碼 Lustre 檔案系統的高效能檔案系統。FSx 當指定`fsx_settings`設定並參考 `[fsx]`區段時，會使用 for Lustre。2.2.1 版中 AWS ParallelCluster 新增FSx了對 Lustre 的支援。

如需 FSx Lustre 的詳細資訊，請參閱 https://aws.amazon.com/fsx/lustre/ 和 https://docs.aws.amazon.com/fsx/。

AWS Identity and Access Management

AWS Identity and Access Management（IAM）在 中用於EC2為 Amazon 為每個個別叢集特定的執行個體 AWS ParallelCluster 提供最低權限IAM角色。 AWS ParallelCluster 執行個體只能存取部署和管理叢集所需的特定API呼叫。

使用 AWS Batch 叢集時，也會為建立叢集時涉及 Docker 映像建置程序的元件建立IAM角色。這些元件包含 Lambda 函數，允許在 Amazon ECR儲存庫之間新增和刪除 Docker 映像。它們也包含允許刪

除為叢集和 CodeBuild 專案建立之 Amazon S3 儲存貯體的函數。資源、 AWS Batch 執行個體和任務也有角色。

如需 的詳細資訊IAM，請參閱 https://aws.amazon.com/iam/和 https://docs.aws.amazon.com/iam/。

AWS Lambda

AWS Lambda（Lambda）會執行協調建立 Docker 映像的函數。Lambda 也會管理自訂叢集資源的清除，例如存放在 Amazon ECR儲存庫和 Amazon S3 上的 Docker 映像。

如需 Lambda 的詳細資訊，請參閱 https://aws.amazon.com/lambda/和 https://docs.aws.amazon.com/lambda/。

Amazon DCV

Amazon DCV 是一種高效能遠端顯示通訊協定，提供在不同的網路條件下，將遠端桌面和應用程式串流交付至任何裝置的安全方法。指定`dcv_settings`設定時，DCV會使用 Amazon，並參考 `[dcv]`一節。Amazon 的支援DCV已新增至 2.5.0 AWS ParallelCluster 版。

如需 Amazon 的詳細資訊DCV，請參閱 https://aws.amazon.com/hpc/dcv/ 和 https://docs.aws.amazon.com/dcv/。

Amazon Route 53

Amazon Route 53（Route 53）用於建立託管區域，其中包含每個運算節點的主機名稱和完整網域名稱。

如需 Route 53 的詳細資訊，請參閱 https://aws.amazon.com/route53/和 https://docs.aws.amazon.com/route53/。

Amazon Simple Notification Service

> **Note**
>
> 本節僅適用於 2 AWS ParallelCluster .11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 Amazon Simple Notification Service。

Amazon Simple Notification Service（Amazon SNS）會收到 Auto Scaling 的通知。這些事件稱為生命週期事件，會在執行個體在 Auto Scaling 群組中啟動或終止時產生。在 中 AWS ParallelCluster，Auto Scaling 群組的 Amazon SNS主題會訂閱 Amazon SQS佇列。

Amazon SNS 不會與 AWS Batch 叢集搭配使用。

如需 Amazon 的詳細資訊SNS，請參閱 https://aws.amazon.com/sns/和 https://docs.aws.amazon.com/sns/。

Amazon Simple Queue Service

> **Note**
>
> 本節僅適用於 2 AWS ParallelCluster .11.4 及 之前的版本。從 2.11.5 版開始， AWS ParallelCluster 不支援使用 Amazon Simple Queue Service。

Amazon Simple Queue Service（Amazon SQS）會保留從 Auto Scaling 傳送的通知、透過 Amazon 傳送的通知SNS，以及從運算節點傳送的通知。Amazon 會將通知傳送與接收通知SQS分離。這可讓主機節點透過輪詢程序處理通知。在此程序中，主機節點會執行 Amazon SQSwatcher並輪詢佇列。Auto Scaling 和運算節點會將訊息發佈到佇列。

Amazon SQS 不會與 AWS Batch 叢集搭配使用。

如需 Amazon 的詳細資訊SQS，請參閱 https://aws.amazon.com/sqs/和 https://docs.aws.amazon.com/sqs/。

Amazon Simple Storage Service

Amazon Simple Storage Service（Amazon S3）會儲存位於每個 中的 AWS ParallelCluster 範本 AWS 區域。 AWS ParallelCluster 可以設定為允許 CLI/SDK 工具使用 Amazon S3。

當您使用 AWS Batch 叢集時，帳戶中的 Amazon S3 儲存貯體會用來儲存相關資料。例如，儲存貯體會存放從提交的任務建立 Docker 映像和指令碼時建立的成品。

如需詳細資訊，請參閱 https://aws.amazon.com/s3/和 https://docs.aws.amazon.com/s3/。

Amazon VPC

Amazon VPC定義叢集中節點所使用的網路。叢集VPC的設定定義於 `[vpc]`一節。

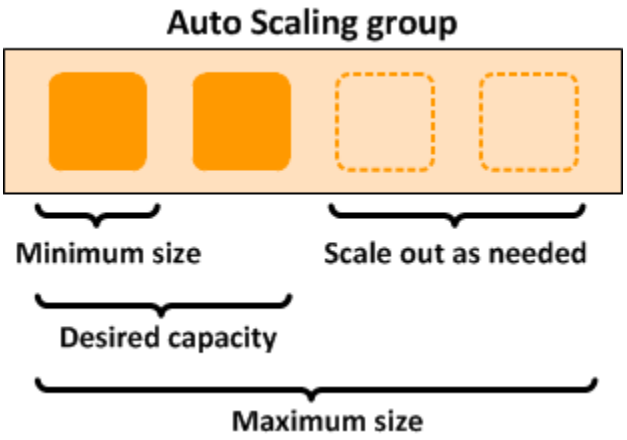
如需 Amazon 的詳細資訊VPC，請參閱 https://aws.amazon.com/vpc/和 https://docs.aws.amazon.com/vpc/。

AWS ParallelCluster Auto Scaling

> **Note**
>
> 本節僅適用於包括 2.11.4 版AWS ParallelCluster本的版本。從版本 2.11.5 開始，AWS ParallelCluster不支持使用SGE或Torque調度程序。您可以在 2.11.4 以下版本中繼續使用它們，但它們不符合AWS服務和支持團隊的未來更新或疑難排解AWS支持的資格。
> 從 2.9.0 AWS ParallelCluster 版開始，自動縮放不支援與 Slurm Workload Manager (Slurm) 搭配使用。若要瞭解Slurm和多重佇列調整規模，請參閱多隊列模式教程。

本主題所述的自動調整規模策略適用於使用 Son of Grid Engine (SGE) 或 Torque Resource Manager (Torque) 部署的 HPC 叢集。使用其中一個排程器部署時，透過管理計算節點的 Auto Scaling 群組，然後視需要變更排程器組態來AWS ParallelCluster實作擴展功能。對於以下項目為基礎的 HPC 叢集AWS Batch，需AWS ParallelCluster仰賴AWS受管理工作排程器提供的彈性調整功能。如需詳細資訊，請參閱 Amazon EC2 自動擴展使用者指南中的什麼是 Amazon EC2 自動擴展規模。

AWS ParallelCluster 部署的叢集在數種方面來說都是彈性的。設定會`initial_queue_size`指定「ComputeFleet自動調整比例」群組的最小大小值，以及所需的容量值。設`max_queue_size`定指定「ComputeFleet自動縮放」群組的最大大小值。



向上縮放

每分鐘，在頭節點上`jobwatcher`運行一個名為的進程。它會評估佇列中待處理任務目前所需的執行個體數目。如果忙碌節點和要求節點的總數大於 Auto Scaling 群組中目前所需的值，則會新增更多執行個體。如果您提交更多工作，則會重新評估佇列並更新 Auto Scaling 群組，直到指定的範`max_queue_size`圍為止。

使用 SGE 排程器，每個任務需要一些插槽才能執行 (一個插槽對應一個處理單位，例如，vCPU)。若要評估處理目前待處理任務所需的執行個體數目，jobwatcher 會將請求的插槽總數除以單一運算節點的容量。與可用 vCPU 數量對應的運算節點容量取決於叢集組態中指定的 Amazon EC2 執行個體類型。

使用Slurm（AWS ParallelCluster版本 2.9.0 之前）和Torque調度程序，根據情況，每個作業可能需要多個節點和每個節點的插槽數量。對於每個請求，jobwatcher 會判斷滿足新運算需求所需的運算節點數目。例如，假設有一個叢集具有 c5.2xlarge (8 vCPU) 做為運算執行個體類型，且有三個排入佇列的待處理任務具有下列需求：

- 任務 1：2 個節點/每個節點 4 個插槽
- 任務 2：3 個節點/每個節點 2 個插槽
- 任務 3：1 個節點/每個節點 4 個插槽

在此範例中，Auto Scaling 群組中jobwatcher需要三個新的運算執行個體才能提供這三項工作。

當前限制：自動擴展邏輯不會考慮部分加載忙碌節點。例如，即使有空插槽，正在執行工作的節點也會被視為忙碌中。

縮小比例

在每個運算節點上，稱為 nodewatcher 的程序會執行，並評估節點的閒置時間。滿足以下兩個條件時，即會終止執行個體：

- 執行個體沒有任務的時段超過 scaledown_idletime (預設設定為 10 分鐘)
- 叢集中沒有待處理任務

若要終止執行個體，請nodewatcher呼叫 TerminateInstanceInAutoScalingGroupAPI 作業，如果 Auto Scaling 群組的大小至少為 Auto Scaling 群組大小下限，則會移除執行個體。此程序會縮減叢集，而不會影響執行中任務。它也會啟用具有固定基礎執行個體數目的彈性叢集。

靜態叢集

對於 HPC，自動擴展的值與任何其他工作負載相同。唯一的差別是，AWS ParallelCluster 具備的程式碼可讓它的互動方式更聰明。例如，如果需要靜態叢集，您可以將initial_queue_size和max_queue_size參數設定為所需叢集的確切大小。，然後將maintain_initial_size參數設定為 true。這會導致「ComputeFleet自動調整」群組的最小、最大容量和所需容量具有相同的值。

教學課程

以下教學課程說明如何開始使用 AWS ParallelCluster，並提供一些常見任務的最佳實務指導。

主題

- 在 AWS ParallelCluster 上執行您的第一個任務
- 建置自訂 AWS ParallelCluster AMI
- 使用AWS ParallelCluster和awsbatch排程器執行 MPI 工作
- 使用自訂 KMS 金鑰進行磁碟加密
- 多隊列模式教程

在 AWS ParallelCluster 上執行您的第一個任務

本教學課程會逐步解說如何在 AWS ParallelCluster 上執行您的第一個 Hello World 任務。

先決條件

- AWS ParallelCluster已安裝。
- AWS CLI已安裝並設定。
- 您有一個 EC2 密鑰對。
- 您擁有具有執行 `pcluster` CLI 所需權限的 IAM 角色。

確認安裝

首先，我們會驗證是否已正確安裝及設定 AWS ParallelCluster。

```
$ pcluster version
```

這會傳回 AWS ParallelCluster 的執行版本。如果輸出提供有關配置的消息，則需要運行以下命令來配置AWS ParallelCluster：

```
$ pcluster configure
```

建立您的第一個叢集

現在要建立您的第一個叢集。由於本教學課程的工作負載不是效能密集的工作負載，我們可以使用 t2.micro 的預設執行個體大小。(對於生產工作負載，您可以選擇最適合您需求的執行個體大小。)

讓我們呼叫您的叢集 hello-world。

```
$ pcluster create hello-world
```

建立叢集後，您會看到類似如下的輸出：

```
Starting: hello-world
Status: parallelcluster-hello-world - CREATE_COMPLETE
MasterPublicIP = 54.148.x.x
ClusterUser: ec2-user
MasterPrivateIP = 192.168.x.x
GangliaPrivateURL = http://192.168.x.x/ganglia/
GangliaPublicURL = http://54.148.x.x/ganglia/
```

訊息 CREATE_COMPLETE 顯示已成功建立叢集。輸出還為我們提供了頭節點的公共和私有 IP 地址。我們需要此 IP 才能登入。

登錄到您的頭節點

使用您的 OpenSSH PEM 文件登錄到您的頭節點。

```
pcluster ssh hello-world -i /path/to/keyfile.pem
```

登入之後，執行命令 qhost 來驗證您的運算節點是否已設置和設定。

```
$ qhost
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR  LOAD  MEMTOT  MEMUSE  SWAPTO
 SWAPUS
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -
      -
ip-192-168-1-125        lx-amd64        2    1    2    2  0.15    3.7G  130.8M 1024.0M
    0.0
ip-192-168-1-126        lx-amd64        2    1    2    2  0.15    3.7G  130.8M 1024.0M
    0.0
```

輸出顯示我們的叢集中有兩個運算節點，兩者都有兩個可供它們使用的執行緒。

使用 SGE 執行您的第一個任務

> **Note**
>
> 此範例僅適用於包含 2.11.4 版AWS ParallelCluster本的版本。從版本 2.11.5 開始，AWS ParallelCluster不支援使用SGE或Torque排程器。

接著，我們建立一個任務，它會休眠一會兒，然後輸出其自己的主機名稱。

建立稱為 `hellojob.sh` 的檔案，其中具有以下內容。

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

接著，使用 `qsub` 來提交任務，並驗證它是否執行。

```
$ qsub hellojob.sh
Your job 1 ("hellojob.sh") has been submitted
```

現在，您可以檢視佇列並檢查此任務的狀態。

```
$ qstat
job-ID  prior   name       user         state submit/start at     queue
          slots ja-task-ID
-----------------------------------------------------------------------------------------------
      1 0.55500 hellojob.s ec2-user     r     03/24/2015 22:23:48
 all.q@ip-192-168-1-125.us-west     1
```

輸出顯示任務目前處於執行中狀態。等待 30 秒讓任務完成，然後再次執行 `qstat`。

```
$ qstat
$
```

現在佇列中沒有任何任務，我們可以在目前的目錄中檢查輸出。

```
$ ls -l
total 8
```

```
-rw-rw-r-- 1 ec2-user ec2-user 48 Mar 24 22:34 hellojob.sh
-rw-r--r-- 1 ec2-user ec2-user  0 Mar 24 22:34 hellojob.sh.e1
-rw-r--r-- 1 ec2-user ec2-user 34 Mar 24 22:34 hellojob.sh.o1
```

在輸出中，我們看到 “e1” 和 “o1” 檔案出現在任務指令碼中。由於該e1文件是空的，所以沒有輸出到 stderr。如果檢視 o1 檔案，我們可以看到來自任務的輸出。

```
$ cat hellojob.sh.o1
Hello World from ip-192-168-1-125
```

輸出也會顯示我們的任務已在執行個體 ip-192-168-1-125 上成功執行。

若要進一步瞭解如何建立和使用叢集，請參閱最佳實務。

建置自訂 AWS ParallelCluster AMI

> **Important**
>
> 我們不建議建立自訂 AMI 作為自訂的方法AWS ParallelCluster。
> 這是因為，在您建立自己的 AMI 之後，您將不再收到更新或錯誤修正與未來版本的AWS ParallelCluster. 此外，如果您構建自定義 AMI，則必須在每個新AWS ParallelCluster版本中重複用於創建自定義 AMI 的步驟。

在進一步閱讀之前，我們建議您先查看「自訂啟動程序動作」一節，以確定您想要進行的修改是否可以在未來的AWS ParallelCluster版本中編寫腳本並支持。

即使構建自定義 AMI 並不理想（因為前面提到的原因），仍然有一些情況下構建自定義 AMI AWS ParallelCluster 是必要的。本教學課程會引導您完成針對這些案例建立自訂 AMI 的程序。

> **Note**
>
> 從AWS ParallelCluster版本 2.6.1 開始，啟動節點時，默認情況下會跳過大多數安裝配方。這是為了改善啟動時間。要以犧牲啟動時間為代價運行所有安裝配方`"skip_install_recipes" : "no"`以獲得更好的向後兼容性，請在`extra_json`設置中添加`cluster`密鑰。例如：
>
> ```
> extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
> ```

先決條件

- AWS ParallelCluster已安裝。
- AWS CLI已安裝並設定。
- 您有一個 EC2 密鑰對。
- 您擁有具有執行 `pcluster` CLI 所需權限的 IAM 角色。

如何自訂 AWS ParallelCluster AMI

有三種使用自定義 AWS ParallelCluster AMI 的方法在下一節中描述。這三種方法中的兩種需要您構建一個新的 AMI，該 AMI 可用於AWS 帳戶. 第三種方法（在運行時使用自定義 AMI）不需要提前構建任何東西，但確實會增加部署的風險。選擇最適合您需求的方法。

修改 AMI

這是最安全，最推薦的方法。由於AWS ParallelCluster基礎 AWS ParallelCluster AMI 通常會隨新版本進行更新，因此此 AMI 具有在安裝和設定時運作所需的所有元件。您可以開始以此做為基礎。

New EC2 console

1. 在 AWS ParallelCluster AMI 列表中，找到與您使用的具體對應AWS 區域的 AMI。您選擇的 AMI 列表必須與您使用的版本AWS ParallelCluster相匹配。執行 `pcluster version` 以驗證版本。對於AWS ParallelCluster版本 2.11.9，請轉到 https://github.com/aws/aws-parallelcluster/blob/v2.11.9/amis.txt。若要選取其他版本，請使用相同的連結，選擇 [標籤:2.11.9] 按鈕，選取 [標籤] 索引標籤，然後選取適當的版本。
2. 請登入 AWS Management Console，並在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
3. 在 Amazon EC2 儀表板中，選擇啟動執行個體。
4. 在應用程式和作業系統映像中，選擇瀏覽更多 AM I，導覽至社群 AM I，然後在搜尋方塊中輸 AWS 區域入您的 AWS ParallelCluster AMI ID。
5. 選擇 AMI，選擇您的實例類型和屬性，選擇您的密鑰對，然後啟動實例。
6. 使用作業系統使用者和 SSH 金鑰來登入執行個體。如需詳細資訊，請瀏覽至執行個體，選取新執行個體，然後連線。
7. 視需要自訂執行個體。
8. 執行下列命令，以備妥執行個體來建立 AMI：

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. 瀏覽至執行個體，選擇新的執行處理，選取執行個體狀態和停止執行處理。
10. 使用 EC2 主控台或建立映像檔從執行個體建AWS CLI立新 AMI。

    從 EC2 主控台

    a. 在導覽窗格中，選擇 Instances (執行個體)。

    b. 選擇您建立和修改的執行個體。

    c. 在動作中，選擇映像和範本，然後選擇建立映像。

    d. 選擇 Create Image (建立映像)。
11. 在叢集配置的自訂欄位中輸入新的 AMI 識別碼。

Old EC2 console

1. 在 AWS ParallelCluster AMI 列表中，找到與您使用的具體對應AWS 區域的 AMI。您選擇的 AMI 列表必須與您使用的版本AWS ParallelCluster相匹配。執行 `pcluster version` 以驗證版本。對於AWS ParallelCluster版本 2.11.9，請轉到 https://github.com/aws/aws-parallelcluster/blob/v2.11.9/amis.txt。若要選取其他版本，請使用相同的連結，選擇 [標籤:2.11.9] 按鈕，選取 [標籤] 索引標籤，然後選取適當的版本。
2. 請登入 AWS Management Console，並在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
3. 在 Amazon EC2 儀表板中，選擇啟動執行個體。
4. 選擇社區 AM I，搜索 AWS ParallelCluster AMI ID，然後選擇它。
5. 選擇您的執行個體類型，然後選取 [下一步：設定執行個體詳細資訊]，或 [檢閱並啟動] 以啟動
6. 選擇 [啟動]，選取您的金鑰配對，然後啟動執行個體。
7. 使用作業系統使用者和 SSH 金鑰來登入執行個體。如需詳細資訊，請瀏覽至執行個體，選取新執行個體，然後連線。
8. 視需要自訂執行個體。
9. 執行下列命令，以備妥執行個體來建立 AMI：

```
sudo /usr/local/sbin/ami_cleanup.sh
```

10. 導覽至執行處理，選擇新執行處理，選取執行處理狀態，然後停止

11.使用 EC2 主控台或建立映像檔從執行個體建AWS CLI立新 AMI。

從 EC2 主控台

a. 在導覽窗格中，選擇 Instances (執行個體)。

b. 選擇您建立和修改的執行個體。

c. 在動作中，選擇影像，然後選擇建立映像。

d. 選擇 Create Image (建立映像)。

12.在叢集配置的自訂欄位中輸入新的 AMI 識別碼。

建置自訂 AWS ParallelCluster AMI

如果已有適當的自訂 AMI 和軟體，您可以在它上面套用 AWS ParallelCluster 所需的變更。

1. 將下列項目與 AWS ParallelCluster CLI 一起安裝在您的本機系統中：
   - Packer：從 Packer 網站中，尋找您作業系統的最新版本並加以安裝。該版本必須至少為 1.4.0，但建議使用最新版本。驗證該`packer`命令在您的 PATH 中可用。

   > **Note**
   >
   > 在AWS ParallelCluster版本 2.8.0 之前，需要使用伯克架（使用安裝`gem install berkshelf`）。`pcluster createami`

2. 設定您的AWS 帳戶認證，以便封裝程式可以代表您呼叫 AWS API 作業。封包器工作所需的最小一組必要許可，記錄在封裝程式文件中 Amazon AMI Builder 主題的 IAM 任務或執行個體角色部分。
3. 使用 `createami` CLI 中的 AWS ParallelCluster，以從您提供的基礎 AMI 來開始建置 AWS ParallelCluster AMI：

   ```
   pcluster createami --ami-id <BASE_AMI> --os <BASE_AMI_OS>
   ```

   > **Important**
   >
   > 您不應該使用來自正在運行的集群中的 AWS ParallelCluster AMI 作<BASE_AMI>為`createami`命令。否則，指令會失敗。

   如需其他參數，請參閱`pcluster createami`。

4. 步驟 4 中的命令會執行封裝程式，它會特別執行下列作業：
   a. 使用提供的基礎 AMI 來啟動執行個體。
   b. 將AWS ParallelCluster食譜套用至執行個體，以安裝相關軟體並執行其他必要的設定工作。
   c. 停止執行個體。
   d. 從執行個體建立新的 AMI。
   e. 在建立 AMI 之後終止執行個體。
   f. 輸出新的 AMI ID 字串以用來建立叢集。
5. 若要建立叢集，請在叢集組態的 custom_ami 欄位中輸入 AMI ID。

> **Note**
> 用來建立自訂 AWS ParallelCluster AMI 的執行個體類型為`t2.xlarge`。此執行個體類型不符合AWS免費方案的資格，因此您需支付建立此 AMI 時建立的任何執行個體費用。

在執行時間使用自訂 AMI

> **Warning**
> 為了避免使用與不兼容的 AMI 的風險AWS ParallelCluster，我們建議您避免使用此方法。在執行階段使用潛在未經測試的 AMI 啟動運算節點時，與所需軟體的AWS ParallelCluster執行階段安裝不相容可能會導致停止運作。AWS ParallelCluster

如果您不想事先創建任何內容，則可以使用 AMI 並AWS ParallelCluster從該 AMI 創建一個。

使用此方法，建立的時間需AWS ParallelCluster要更長的時間，因為必須安裝建立叢集AWS ParallelCluster時所需的所有軟體。此外，擴展也需要更長的時間。

- 在叢集組態的 custom_ami 欄位中輸入 AMI id。

使用AWS ParallelCluster和`awsbatch`排程器執行 MPI 工作

本教學課程會逐步解說如何搭配 `awsbatch` 做為排程器來執行 MPI 任務。

先決條件

- AWS ParallelCluster已安裝。
- AWS CLI已安裝並設定。
- 您有一個 EC2 密鑰對。
- 您擁有具有執行 `pcluster` CLI 所需權限的 IAM 角色。

建立叢集

首先，讓我們為叢集建立一個組態，使用 awsbatch 做為排程器。務必以您在設定時所建立的資源插入 vpc 區段和 key_name 欄位中缺少的資料。

```
[global]
sanity_check = true

[aws]
aws_region_name = us-east-1

[cluster awsbatch]
base_os = alinux
# Replace with the name of the key you intend to use.
key_name = key-#######
vpc_settings = my-vpc
scheduler = awsbatch
compute_instance_type = optimal
min_vcpus = 2
desired_vcpus = 2
max_vcpus = 24

[vpc my-vpc]
# Replace with the id of the vpc you intend to use.
vpc_id = vpc-#######
# Replace with id of the subnet for the Head node.
master_subnet_id = subnet-#######
# Replace with id of the subnet for the Compute nodes.
# A NAT Gateway is required for MNP.
compute_subnet_id = subnet-#######
```

您現在可以開始建立叢集。讓我們呼叫叢集 *awsbatch-tutorial*。

```
$ pcluster create -c /path/to/the/created/config/aws_batch.config -t awsbatch awsbatch-tutorial
```

建立叢集後，您會看到類似如下的輸出：

```
Beginning cluster creation for cluster: awsbatch-tutorial
Creating stack named: parallelcluster-awsbatch
Status: parallelcluster-awsbatch - CREATE_COMPLETE
MasterPublicIP: 54.160.xxx.xxx
ClusterUser: ec2-user
MasterPrivateIP: 10.0.0.15
```

登錄到您的頭節點

AWS ParallelCluster 批次 CLI 命令都可在安裝 AWS ParallelCluster 的用戶端機器上使用。但是，我們將 SSH 進入頭節點並從那裡提交作業。這讓我們可以利用磁頭和所有執行AWS Batch作業的 Docker 執行個體之間共用的 NFS 磁碟區。

使用您的 SSH pem 檔案登入您的頭節點。

```
$ pcluster ssh awsbatch-tutorial -i /path/to/keyfile.pem
```

登入後，執行命令`awsbqueues`並顯示已設定`awsbhosts`的AWS Batch佇列和執行中的 Amazon ECS 執行個體。

```
[ec2-user@ip-10-0-0-111 ~]$ awsbqueues
jobQueueName                      status
--------------------------------  --------
parallelcluster-awsbatch-tutorial  VALID

[ec2-user@ip-10-0-0-111 ~]$ awsbhosts
ec2InstanceId        instanceType    privateIpAddress    publicIpAddress
 runningJobs
-------------------  --------------  ------------------  -----------------
 -------------
i-0d6a0c8c560cd5bed  m4.large        10.0.0.235          34.239.174.236
 0
```

如您在輸出中所見，我們只有一個執行中的主機。這是由於我們在組態中針對 `min_vcpus` 選擇的值所致。如果您想要顯示有關 AWS Batch 佇列和主機的其他詳細資訊，請將 `-d` 旗標新增至命令。

使用 AWS Batch 執行您的第一個任務

在移至 MPI 之前，讓我們先建立一個虛擬任務，它會先休眠一會兒，之後便會輸出其自己的主機名稱，並問候以參數傳遞的名稱。

建立名為 "hellojob.sh" 的檔案，其中內容如下。

```
#!/bin/bash

sleep 30
echo "Hello $1 from $HOSTNAME"
echo "Hello $1 from $HOSTNAME" > "/shared/secret_message_for_${1}_by_
${AWS_BATCH_JOB_ID}"
```

接著，使用 awsbsub 來提交任務，並驗證它是否執行。

```
$ awsbsub -jn hello -cf hellojob.sh Luca
Job 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2 (hello) has been submitted.
```

檢視佇列並檢查任務的狀態。

```
$ awsbstat
jobId                                    jobName      status    startedAt
 stoppedAt    exitCode
-------------------------------------  -----------  --------  -------------------
 -----------  ----------
6efe6c7c-4943-4c1a-baf5-edbfeccab5d2  hello        RUNNING   2018-11-12 09:41:29  -
        -
```

輸出提供任務的詳細資訊。

```
$ awsbstat 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
jobId                    : 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
jobName                  : hello
createdAt                : 2018-11-12 09:41:21
startedAt                : 2018-11-12 09:41:29
stoppedAt                : -
status                   : RUNNING
statusReason             : -
jobDefinition            : parallelcluster-myBatch:1
jobQueue                 : parallelcluster-myBatch
```

```
command                   : /bin/bash -c 'aws s3 --region us-east-1 cp s3://
parallelcluster-mybatch-lui1ftboklhpns95/batch/job-hellojob_sh-1542015680924.sh /
tmp/batch/job-hellojob_sh-1542015680924.sh; bash /tmp/batch/job-
hellojob_sh-1542015680924.sh Luca'
exitCode                  : -
reason                    : -
vcpus                     : 1
memory[MB]                : 128
nodes                     : 1
logStream                 : parallelcluster-myBatch/default/c75dac4a-5aca-4238-
a4dd-078037453554
log                       : https://console.aws.amazon.com/cloudwatch/home?region=us-
east-1#logEventViewer:group=/aws/batch/job;stream=parallelcluster-myBatch/default/
c75dac4a-5aca-4238-a4dd-078037453554
-------------------------
```

請注意，任務目前處於 RUNNING 狀態。等待 30 秒讓任務完成，然後再次執行 awsbstat。

```
$ awsbstat
jobId                                    jobName      status    startedAt
 stoppedAt    exitCode
-------------------------------------  -----------  --------  -------------------
 -----------  ----------
```

現在，您可以看到任務處於 SUCCEEDED 狀態。

```
$ awsbstat -s SUCCEEDED
jobId                                    jobName      status     startedAt
 stoppedAt              exitCode
-------------------------------------  -----------  ---------  -------------------
 -------------------  ----------
6efe6c7c-4943-4c1a-baf5-edbfeccab5d2  hello        SUCCEEDED  2018-11-12 09:41:29
 2018-11-12 09:42:00           0
```

由於現在佇列中沒有任何任務，我們可以透過 awsbout 命令來檢查輸出。

```
$ awsbout 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
2018-11-12 09:41:29: Starting Job 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
download: s3://parallelcluster-mybatch-lui1ftboklhpns95/batch/job-
hellojob_sh-1542015680924.sh to tmp/batch/job-hellojob_sh-1542015680924.sh
2018-11-12 09:42:00: Hello Luca from ip-172-31-4-234
```

我們可以看到任務已在執行個體 "ip-172-31-4-234" 上成功執行。

如果您查看 /shared 目錄，您會發現一則給您的秘密訊息。

若要探索不是本教學課程一部分的所有可用功能，請參閱 AWS ParallelCluster 批次 CLI 文件。當您準備好繼續教學課程時，讓我們繼續看如何提交 MPI 任務。

在多節點平行環境中執行 MPI 任務

在仍然登錄到 head 節點的同時，在名為的/shared目錄中創建一個文件mpi_hello_world.c。將以下 MPI 程式新增至這個檔案：

```
// Copyright 2011 www.mpitutorial.com
//
// An intro MPI hello world program that uses MPI_Init, MPI_Comm_size,
// MPI_Comm_rank, MPI_Finalize, and MPI_Get_processor_name.
//
#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();
}
```

現在，將下列程式碼儲存為 submit_mpi.sh：

```
#!/bin/bash
echo "ip container: $(/sbin/ip -o -4 addr list eth0 | awk '{print $4}' | cut -d/ -f1)"
echo "ip host: $(curl -s "http://169.254.169.254/latest/meta-data/local-ipv4")"

# get shared dir
IFS=',' _shared_dirs=(${PCLUSTER_SHARED_DIRS})
_shared_dir=${_shared_dirs[0]}
_job_dir="${_shared_dir}/${AWS_BATCH_JOB_ID%#*}-${AWS_BATCH_JOB_ATTEMPT}"
_exit_code_file="${_job_dir}/batch-exit-code"

if [[ "${AWS_BATCH_JOB_NODE_INDEX}" -eq  "${AWS_BATCH_JOB_MAIN_NODE_INDEX}" ]]; then
    echo "Hello I'm the main node $HOSTNAME! I run the mpi job!"

    mkdir -p "${_job_dir}"

    echo "Compiling..."
    /usr/lib64/openmpi/bin/mpicc -o "${_job_dir}/mpi_hello_world" "${_shared_dir}/
mpi_hello_world.c"

    echo "Running..."
    /usr/lib64/openmpi/bin/mpirun --mca btl_tcp_if_include eth0 --allow-run-as-root --
machinefile "${HOME}/hostfile" "${_job_dir}/mpi_hello_world"

    # Write exit status code
    echo "0" > "${_exit_code_file}"
    # Waiting for compute nodes to terminate
    sleep 30
else
    echo "Hello I'm the compute node $HOSTNAME! I let the main node orchestrate the mpi
 processing!"
    # Since mpi orchestration happens on the main node, we need to make sure the
 containers representing the compute
    # nodes are not terminated. A simple trick is to wait for a file containing the
 status code to be created.
    # All compute nodes are terminated by AWS Batch if the main node exits abruptly.
    while [ ! -f "${_exit_code_file}" ]; do
```

```
        sleep 2
    done
    exit $(cat "${_exit_code_file}")
fi
```

我們現在準備好提交第一個 MPI 任務，讓它在三個節點上同時執行：

```
$ awsbsub -n 3 -cf submit_mpi.sh
```

現在，讓我們監控任務狀態，等待它進入 RUNNING 狀態：

```
$ watch awsbstat -d
```

當任務進入 RUNNING 狀態時，我們可以查看其輸出。若要顯示主節點的輸出，請將 #0 附加到任務 ID。若要顯示運算節點的輸出，請使用 #1 和 #2：

```
[ec2-user@ip-10-0-0-111 ~]$ awsbout -s 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#0
2018-11-27 15:50:10: Job id: 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#0
2018-11-27 15:50:10: Initializing the environment...
2018-11-27 15:50:10: Starting ssh agents...
2018-11-27 15:50:11: Agent pid 7
2018-11-27 15:50:11: Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
2018-11-27 15:50:11: Mounting shared file system...
2018-11-27 15:50:11: Generating hostfile...
2018-11-27 15:50:11: Detected 1/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:50:26: Detected 1/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:50:41: Detected 1/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:50:56: Detected 3/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:51:11: Starting the job...
download: s3://parallelcluster-awsbatch-tutorial-iwyl4458saiwgwvg/batch/job-
submit_mpi_sh-1543333713772.sh to tmp/batch/job-submit_mpi_sh-1543333713772.sh
2018-11-27 15:51:12: ip container: 10.0.0.180
2018-11-27 15:51:12: ip host: 10.0.0.245
2018-11-27 15:51:12: Compiling...
2018-11-27 15:51:12: Running...
2018-11-27 15:51:12: Hello I'm the main node! I run the mpi job!
2018-11-27 15:51:12: Warning: Permanently added '10.0.0.199' (RSA) to the list of known
 hosts.
```

```
2018-11-27 15:51:12: Warning: Permanently added '10.0.0.147' (RSA) to the list of known
 hosts.
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-180.ec2.internal, rank 1 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-199.ec2.internal, rank 5 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-180.ec2.internal, rank 0 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-199.ec2.internal, rank 4 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-147.ec2.internal, rank 2 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-147.ec2.internal, rank 3 out
 of 6 processors

[ec2-user@ip-10-0-0-111 ~]$ awsbout -s 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#1
2018-11-27 15:50:52: Job id: 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#1
2018-11-27 15:50:52: Initializing the environment...
2018-11-27 15:50:52: Starting ssh agents...
2018-11-27 15:50:52: Agent pid 7
2018-11-27 15:50:52: Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
2018-11-27 15:50:52: Mounting shared file system...
2018-11-27 15:50:52: Generating hostfile...
2018-11-27 15:50:52: Starting the job...
download: s3://parallelcluster-awsbatch-tutorial-iwyl4458saiwgwvg/batch/job-
submit_mpi_sh-1543333713772.sh to tmp/batch/job-submit_mpi_sh-1543333713772.sh
2018-11-27 15:50:53: ip container: 10.0.0.199
2018-11-27 15:50:53: ip host: 10.0.0.227
2018-11-27 15:50:53: Compiling...
2018-11-27 15:50:53: Running...
2018-11-27 15:50:53: Hello I'm a compute node! I let the main node orchestrate the mpi
 execution!
```

我們現在可以確認任務已成功完成：

```
[ec2-user@ip-10-0-0-111 ~]$ awsbstat -s ALL
jobId                                    jobName        status     startedAt
 stoppedAt            exitCode
-------------------------------------  -------------  ---------  -------------------
 -------------------  ----------
5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d  submit_mpi_sh  SUCCEEDED  2018-11-27 15:50:10
 2018-11-27 15:51:26  -
```

注意：如果想要在任務結束之前終止任務，您可以使用 `awsbkill` 命令。

使用自訂 KMS 金鑰進行磁碟加密

AWS ParallelCluster 支援組態選項 `ebs_kms_key_id` 和 `fsx_kms_key_id`。這些選項可讓您為 Amazon EBS 磁碟加密或 FSx (適用於 Lustre) 提供自訂AWS KMS金鑰。若要使用它們，請指定 `ec2_iam_role`。

若要建立叢集，AWS KMS 金鑰必須知道叢集角色的名稱。這會防止您使用建立叢集時所建立的角色，因此需要自訂 `ec2_iam_role`。

先決條件

- AWS ParallelCluster已安裝。
- AWS CLI已安裝並設定。
- 您有一個 EC2 密鑰對。
- 您擁有具有執行 `pcluster` CLI 所需權限的 IAM 角色。

建立 角色

首先建立政策：

1. 前往身分與存取權管理主控台：https://console.aws.amazon.com/iam/home。
2. 在 Policies (政策) > Create policy (建立政策) 下，按一下 JSON 標籤。
3. 做為政策的內文，貼入 執行個體政策中。務必取代所有出現的 *<AWS ACCOUNT ID>* 和 *<REGION>*。
4. 將政策命名為 `ParallelClusterInstancePolicy`，然後按一下 Create Policy (建立政策)。

接著，建立角色：

1. 在 Roles (角色) 下，建立角色。
2. 按一下 `EC2` 做為信任的實體。
3. 在 Permissions (許可) 下，搜尋您剛建立的 `ParallelClusterInstancePolicy` 角色並連接它。
4. 將角色命名為 `ParallelClusterInstanceRole`，然後按一下 Create Role (建立角色)。

授予您的金鑰權限

在AWS KMS主控台 > 客戶受管金鑰中，按一下金鑰的別名或金鑰 ID。

按一下 [金鑰原則] 索引標籤下方 [金鑰使用者] 方塊中的 [新增] 按鈕，然後搜尋ParallelClusterInstanceRole您剛建立的。連接它。

建立叢集

現在建立叢集。以下是叢集的範例，其中具有加密的 `Raid 0` 磁碟機：

```
[cluster default]
...
raid_settings = rs
ec2_iam_role = ParallelClusterInstanceRole

[raid rs]
shared_dir = raid
raid_type = 0
num_of_raid_volumes = 2
volume_size = 100
encrypted = true
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

以下是使用 Lustre 檔案系統的 FSx 範例：

```
[cluster default]
...
fsx_settings = fs
ec2_iam_role = ParallelClusterInstanceRole

[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
import_path = s3://bucket
weekly_maintenance_start_time = 1:00:00
fsx_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

類似的組態適用於以亞馬遜 EBS 和 Amazon FSx 為基礎的檔案系統。

多隊列模式教程

AWS ParallelCluster以多重佇列模式執行作業

本教程將引導您完成AWS ParallelCluster與上運行您的第一個 Hello World 工作多重佇列模式。

先決條件

- AWS ParallelCluster已安裝。
- AWS CLI已安裝並設定。
- 您有一個 EC2 密鑰對。
- 您擁有具有執行 `pcluster` CLI 所需權限的 IAM 角色。

> **Note**
>
> 僅AWS ParallelCluster版本 2.9.0 或更新版本支援多重佇列模式。

配置您的叢集

首先，請執行下列命令來確認AWS ParallelCluster是否已正確安裝。

```
$ pcluster version
```

如需有關 `pcluster version` 的詳細資訊，請參閱 `pcluster version`。

此命令會傳回的執行中版本AWS ParallelCluster。

接下來，運行`pcluster configure`以生成基本配置文件。遵循此指令的所有提示進行操作。

```
$ pcluster configure
```

如需 `pcluster configure` 命令的詳細資訊，請參閱`pcluster configure`。

完成此步驟之後，您應該在下面有一個基本的組態檔案`~/.parallelcluster/config`。此檔案應包含基本叢集配置和 VPC 區段。

本教學課程的下一部分將說明如何修改新建立的配置，以及如何啟動具有多個佇列的叢集。

> **Note**
>
> 本教學課程中使用的部分執行個體不符合免費方案資格。

在本教學課程中，請使用下列規劃。

```
[global]
update_check = true
sanity_check = true
cluster_template = multi-queue

[aws]
aws_region_name = <Your AWS ##>

[scaling demo]
scaledown_idletime = 5              # optional, defaults to 10 minutes

[cluster multi-queue-special]
key_name = < Your key name >
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge    # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo             # optional, defaults to no custom scaling settings
queue_settings = efa,gpu

[cluster multi-queue]
key_name = <Your SSH key name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge    # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = spot,ondemand

[queue spot]
compute_resource_settings = spot_i1,spot_i2
compute_type = spot                 # optional, defaults to ondemand

[compute_resource spot_i1]
instance_type = c5.xlarge
min_count = 0                       # optional, defaults to 0
```

```
max_count = 10                      # optional, defaults to 10

[compute_resource spot_i2]
instance_type = t2.micro
min_count = 1
initial_count = 2

[queue ondemand]
compute_resource_settings = ondemand_i1
disable_hyperthreading = true       # optional, defaults to false

[compute_resource ondemand_i1]
instance_type = c5.2xlarge
```

建立叢集

本節詳細說明如何建立多重佇列模式叢集。

首先，命名您的叢集multi-queue-hello-world，並根據上一節中定義的multi-queue叢集區段建立叢集。

```
$ pcluster create multi-queue-hello-world -t multi-queue
```

如需有關 pcluster create 的詳細資訊，請參閱 pcluster create。

建立叢集時，會顯示下列輸出：

```
Beginning cluster creation for cluster: multi-queue-hello-world
Creating stack named: parallelcluster-multi-queue-hello-world
Status: parallelcluster-multi-queue-hello-world - CREATE_COMPLETE
MasterPublicIP: 3.130.xxx.xx
ClusterUser: ec2-user
MasterPrivateIP: 172.31.xx.xx
```

此訊息CREATE_COMPLETE表示叢集已成功建立。輸出還提供了頭節點的公共和私有 IP 地址。

登錄到您的頭節點

使用您的私密安全殼層金鑰檔案登入您的頭節點。

```
$ pcluster ssh multi-queue-hello-world -i ~/path/to/keyfile.pem
```

如需有關 pcluster ssh 的詳細資訊，請參閱 pcluster ssh。

登入後，執行sinfo命令以確認您的排程器佇列是否已設定和設定。

如需有關的詳細資訊sinfo，請參閱Slurm文件中的 sinfo。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     18  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      2   idle spot-dy-t2micro-1,spot-st-t2micro-1
```

輸出顯示您的叢集中有兩個處於可用idle狀態的t2.micro運算節點。

> **Note**
>
> - spot-st-t2micro-1是名稱st中的靜態節點。此節點始終可用，並對應於叢集配置min_count = 1中的。
> - spot-dy-t2micro-1是名稱dy中的動態節點。此節點目前可用，因為它會initial_count - min_count = 1根據您的叢集配置對應。自訂scaledown_idletime 5 分鐘後，此節點會縮減。

其他節點全部處於省電狀態，以節點狀態的~尾碼顯示，沒有 EC2 執行個體支援它們。預設佇列的佇列名稱後會以*尾碼指定，預設工作佇列spot也是如此。

在多個佇列模式下執行工作

接下來，嘗試運行工作睡一段時間。工作稍後會輸出自己的主機名稱。確保此腳本可以由當前用戶運行。

```
$ cat hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

使用sbatch指令送出工作。使用此-N 2選項為此工作要求兩個節點，並確認工作是否成功送出。如需有關的詳細資訊sbatch，請參閱 sbatchS lurm 文件中的。

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 2
```

您可以使用squeue指令檢視佇列並檢查工作狀態。請注意，因為您並未指定特定佇列，因此會使用預設 queue (spot)。如需相關資訊squeue，請參閱Slurm文件squeue中的。

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 2      spot     wrap ec2-user  R       0:10      2 spot-dy-
t2micro-1,spot-st-t2micro-1
```

輸出顯示任務目前處於執行中狀態。等待 30 秒讓任務完成，然後再次執行 squeue。

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

現在佇列中的工作都已完成，請在目前目錄slurm-2.out中尋找輸出檔案。

```
$ cat slurm-2.out
Hello World from spot-dy-t2micro-1
Hello World from spot-st-t2micro-1
```

輸出也會顯示我們的工作在spot-st-t2micro-1和spot-st-t2micro-2節點上成功執行。

現在，透過使用下列命令指定特定執行個體的限制來提交相同的工作。

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 3
```

您已將這些參數用於sbatch。

- -N 3-請求三個節點
- -p spot— 將工作提交至spot佇列。您也可以指定將工作提交至ondemand佇列-p ondemand。
- -C "[c5.xlarge*1&t2.micro*2]"— 指定此工作的特定節點限制。這會要求此工作使用一 (1) c5.xlarge 個t2.micro節點和兩 (2) 個節點。

執行sinfo命令以檢視節點和佇列。(中的佇列稱AWS ParallelCluster為中的分割區Slurm。)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite      1   mix# spot-dy-c5xlarge-1
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      2  alloc spot-dy-t2micro-1,spot-st-t2micro-1
```

節點正在開啟電源。這是由節點狀態的#後綴表示。執行命squeue令以檢視叢集中工作的相關資訊。

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 3      spot     wrap ec2-user CF       0:04      3 spot-dy-
c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```

您的工作處於 CF (CONFIGURING) 狀態，等待執行個體擴展並加入叢集。

大約三分鐘後，節點應可供使用，且工作會進入 R (RUNNING) 狀態。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      1    mix spot-dy-c5xlarge-1
spot*        up   infinite      2  alloc spot-dy-t2micro-1,spot-st-t2micro-1
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 3      spot     wrap ec2-user  R       0:04      3 spot-dy-
c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```

工作完成，且所有三個節點都處於idle狀態。

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      3   idle spot-dy-c5xlarge-1,spot-dy-t2micro-1,spot-st-
t2micro-1
```

然後，佇列中沒有剩餘的工作之後，您可以slurm-3.out在本機目錄中檢查。

```
$ cat slurm-3.out
Hello World from spot-dy-c5xlarge-1
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
```

輸出也會顯示工作在對應節點上成功執行。

您可以觀察縮放過程。在您的群集配置中，您指定了 5 分鐘scaledown_idletime的自定義。處於閒置狀態五分鐘後，您的動態節點spot-dy-c5xlarge-1spot-dy-t2micro-1會自動縮小並進入POWER_DOWN模式。請注意，靜態節點spot-st-t2micro-1不會縮小。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite      2  idle% spot-dy-c5xlarge-1,spot-dy-t2micro-1
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      1   idle spot-st-t2micro-1
```

從上面的代碼中，你可以看到，spot-dy-c5xlarge-1並處spot-dy-t2micro-1於POWER_DOWN模式。這是由%後綴表示。對應的執行個體會立即終止，但節點會維持在POWER_DOWN狀態，且無法使用 120 秒 (兩分鐘)。在此時間之後，節點會恢復省電狀態，並可再次使用。如需詳細資訊，請參閱Slurm多個佇列模式的指南。

這應該是叢集的最終狀態：

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     19  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2micro-[1-9]
spot*        up   infinite      1   idle spot-st-t2micro-1
```

登出叢集之後，您可以透過執行來進行清理pcluster delete。若要取得更多資訊 pcluster listpcluster delete，請參閱〈〉pcluster list和〈〉pcluster delete。

```
$ pcluster list
multi-queue CREATE_COMPLETE 2.11.9
$ pcluster delete multi-queue
Deleting: multi-queue
```

```
...
```

使用 EFA 和 GPU 執行個體在叢集上執行工作

本教學課程的這一部分詳細說明如何修改組態，並啟動具有多個佇列的叢集，其中包含具有 EFA 網路和 GPU 資源的執行個體。請注意，本教學課程中使用的執行個體價格較高。

請先檢查您的帳戶限制，確保您已獲得授權使用這些執行個體，然後再繼續執行本教學課程中概述的步驟。

使用下列指令修改組態檔案。

```
[global]
update_check = true
sanity_check = true
cluster_template = multi-queue-special

[aws]
aws_region_name = <Your AWS ##>

[scaling demo]
scaledown_idletime = 5

[cluster multi-queue-special]
key_name = <Your SSH key name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge    # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = efa,gpu

[queue gpu]
compute_resource_settings = gpu_i1
disable_hyperthreading = true       # optional, defaults to false

[compute_resource gpu_i1]
instance_type = g3.8xlarge

[queue efa]
compute_resource_settings = efa_i1
enable_efa = true
placement_group = DYNAMIC           # optional, defaults to no placement group settings
```

```
[compute_resource efa_i1]
instance_type = c5n.18xlarge
max_count = 5
```

建立叢集

```
$ pcluster create multi-queue-special -t multi-queue-special
```

建立叢集之後，請使用您的私密安全殼層金鑰檔案登入您的主節點。

```
$ pcluster ssh multi-queue-special -i ~/path/to/keyfile.pem
```

這應該是叢集的初始狀態：

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      5  idle~ efa-dy-c5n18xlarge-[1-5]
gpu          up   infinite     10  idle~ gpu-dy-g38xlarge-[1-10]
```

本節說明如何提交某些工作以檢查節點是否具有 EFA 或 GPU 資源。

首先，撰寫工作指令碼。 efa_job.sh將睡眠 30 秒。之後，在lspci命令的輸出中尋找 EFA。gpu_job.sh將睡眠 30 秒。之後，運行nvidia-smi以顯示有關節點的 GPU 信息。

```
$ cat efa_job.sh
#!/bin/bash

sleep 30
lspci | grep "EFA"

$ cat gpu_job.sh
#!/bin/bash

sleep 30
nvidia-smi

$ chmod +x efa_job.sh
$ chmod +x gpu_job.sh
```

使用下列方式提交工sbatch作：

```
$ sbatch -p efa --wrap "srun efa_job.sh"
Submitted batch job 2
$ sbatch -p gpu --wrap "srun gpu_job.sh" -G 1
Submitted batch job 3
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 2       efa     wrap ec2-user CF       0:32      1 efa-dy-
c5n18xlarge-1
                 3       gpu     wrap ec2-user CF       0:20      1 gpu-dy-g38xlarge-1
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      1   mix# efa-dy-c5n18xlarge-1
efa*         up   infinite      4  idle~ efa-dy-c5n18xlarge-[2-5]
gpu          up   infinite      1   mix# gpu-dy-g38xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]
```

幾分鐘後，您應該會看到節點線上和工作正在執行。

```
[ec2-user@ip-172-31-15-251 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      4  idle~ efa-dy-c5n18xlarge-[2-5]
efa*         up   infinite      1    mix efa-dy-c5n18xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]
gpu          up   infinite      1    mix gpu-dy-g38xlarge-1
[ec2-user@ip-172-31-15-251 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 4       gpu     wrap ec2-user  R       0:06      1 gpu-dy-g38xlarge-1
                 5       efa     wrap ec2-user  R       0:01      1 efa-dy-
c5n18xlarge-1
```

工作完成後，檢查輸出。從slurm-2.out檔案中的輸出，您可以看到 EFA 存在於efa-dy-c5n18xlarge-1節點上。從slurm-3.out文件中的輸出中，您可以看到nvidia-smi輸出包含gpu-dy-g38xlarge-1節點的 GPU 信息。

```
$ cat slurm-2.out
00:06.0 Ethernet controller: Amazon.com, Inc. Elastic Fabric Adapter (EFA)

$ cat slurm-3.out
Thu Oct  1 22:19:18 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.51.05    Driver Version: 450.51.05    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
```

```
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla M60           Off  | 00000000:00:1D.0 Off |                    0 |
| N/A   28C    P0    38W / 150W |      0MiB /  7618MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla M60           Off  | 00000000:00:1E.0 Off |                    0 |
| N/A   36C    P0    37W / 150W |      0MiB /  7618MiB |     98%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                 GPU Memory |
|        ID   ID                                                  Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

您可以觀察縮放過程。在叢集配置中，您先前指定scaledown_idletime的自訂時間為 5 分鐘。因此，在閒置狀態五分鐘後，您的動態節點會自動縮小並進入POWER_DOWN模式。spot-dy-c5xlarge-1 spot-dy-t2micro-1最後，節點會進入省電模式，並可再次使用。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      1  idle% efa-dy-c5n18xlarge-1
efa*         up   infinite      4  idle~ efa-dy-c5n18xlarge-[2-5]
gpu          up   infinite      1  idle% gpu-dy-g38xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]

# After 120 seconds
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      5  idle~ efa-dy-c5n18xlarge-[1-5]
gpu          up   infinite     10  idle~ gpu-dy-g38xlarge-[1-10]
```

登出叢集之後，您可以透過執行來進行清理pcluster delete *<cluster name>*。

```
$ pcluster list
multi-queue-special CREATE_COMPLETE 2.11.9
```

```
$ pcluster delete multi-queue-special
Deleting: multi-queue-special
...
```

如需詳細資訊，請參閱 Slurm 多個佇列模式的指南。

開發

您可以使用以下區段，開始 AWS ParallelCluster 的開發。

> **Important**
>
> 以下區段包括使用自訂版本之技術指南配方及自訂 AWS ParallelCluster 節點套件的指示。此資訊涵蓋自訂 AWS ParallelCluster 的進階方法，可處理難以除錯的潛在問題。AWS ParallelCluster 團隊強烈建議在自訂引導操作中使用指令碼進行自訂，因為安裝後勾點通常較容易除錯，在 AWS ParallelCluster 的發行過程中也較容易轉移。

主題

設置自定義AWS ParallelCluster食譜

> **Important**
>
> 下列是要使用自訂版本之 AWS ParallelCluster 技術指南配方的指示。這是自訂 AWS ParallelCluster 的進階方法，可處理難以除錯的潛在問題。AWS ParallelCluster 團隊強烈建議在自訂引導操作中使用指令碼進行自訂，因為安裝後勾點通常較容易除錯，在 AWS ParallelCluster 的發行過程中也較容易轉移。

步驟

1. 識別您克隆AWS ParallelCluster食譜代碼的AWS ParallelCluster食譜工作目錄。

```
_cookbookDir=<path to cookbook>
```

2. 偵測 AWS ParallelCluster 技術指南的目前版本。

```
_version=$(grep version ${_cookbookDir}/metadata.rb|awk '{print $2}'| tr -d \')
```

3. 建立 AWS ParallelCluster 技術指南的封存並計算其 md5。

```
cd "${_cookbookDir}"
_stashName=$(git stash create)
git archive --format tar --prefix="aws-parallelcluster-cookbook-${_version}/"
 "${_stashName:-HEAD}" | gzip > "aws-parallelcluster-cookbook-${_version}.tgz"
md5sum "aws-parallelcluster-cookbook-${_version}.tgz" > "aws-parallelcluster-
cookbook-${_version}.md5"
```

4. 建立 Amazon S3 儲存貯體，並將存檔、其 md5 及其上次修改日期上傳至儲存貯體。透過 public-read ACL 來提供可公開讀取的許可。

```
_bucket=<the bucket name>
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.tgz s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.tgz
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.md5 s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.md5
aws s3api head-object --bucket ${_bucket} --key cookbooks/aws-parallelcluster-
cookbook-${_version}.tgz --output text --query LastModified > aws-parallelcluster-
cookbook-${_version}.tgz.date
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.tgz.date s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.tgz.date
```

5. 將下列變數新增至AWS ParallelCluster組態檔案的[cluster]區段下。

```
custom_chef_cookbook = https://${_bucket}.s3.<the bucket region>.amazonaws.com/
cookbooks/aws-parallelcluster-cookbook-${_version}.tgz
extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
```

> **Note**
>
> 從 2.6.1 AWS ParallelCluster 版開始，啟動節點時，預設會略過大部分安裝配方，以縮短啟動時間。要跳過大多數安裝配方以獲得更好的啟動時間，以犧牲向後兼容性為代價，請"skip_install_recipes" : "no"從extra_json設置中的cluster密鑰中刪除。

設置自定義AWS ParallelCluster節點包

> **Warning**
>
> 下列是要使用自訂版本之 AWS ParallelCluster 節點套件的指示。這是自訂 AWS ParallelCluster 的進階方法，可處理難以除錯的潛在問題。AWS ParallelCluster 團隊強烈建議在自訂引導操作中使用指令碼進行自訂，因為安裝後勾點通常較容易除錯，在 AWS ParallelCluster 的發行過程中也較容易轉移。

步驟

1. 識別 AWS ParallelCluster 節點工作目錄，您已在其中複製 AWS ParallelCluster 節點程式碼。

```
_nodeDir=<path to node package>
```

2. 偵測 AWS ParallelCluster 節點的目前版本。

```
_version=$(grep "version = \"" ${_nodeDir}/setup.py |awk '{print $3}' | tr -d \")
```

3. 建立 AWS ParallelCluster 節點的存檔。

```
cd "${_nodeDir}"
_stashName=$(git stash create)
git archive --format tar --prefix="aws-parallelcluster-node-${_version}/"
 "${_stashName:-HEAD}" | gzip > "aws-parallelcluster-node-${_version}.tgz"
```

4. 建立 Amazon S3 儲存貯體，然後將存檔上傳到儲存貯體。透過 public-read ACL 來提供可公開讀取的許可。

```
_bucket=<the bucket name>
aws s3 cp --acl public-read aws-parallelcluster-node-${_version}.tgz s3://${_bucket}/
node/aws-parallelcluster-node-${_version}.tgz
```

5. 將下列變數新增至AWS ParallelCluster組態檔案中的[cluster]區段下。

```
extra_json = { "cluster" : { "custom_node_package" : "https://${_bucket}.s3.<the
 bucket region>.amazonaws.com/node/aws-parallelcluster-node-${_version}.tgz",
 "skip_install_recipes" : "no" } }
```

> **Note**
>
> 從 2.6.1 AWS ParallelCluster 版開始，啟動節點時，預設會略過大部分安裝配方，以縮短啟動時間。要跳過大多數安裝配方以獲得更好的啟動時間，以犧牲向後兼容性為代價，請`"skip_install_recipes" : "no"`從`extra_json`設置中的`cluster`密鑰中刪除。

AWS ParallelCluster 疑難排解

AWS ParallelCluster 社群會維護 Wiki 頁面，該頁面在 AWS ParallelCluster GitHub Wiki 上提供許多疑難排解提示。如需已知問題的清單，請參閱已知問題。

主題

擷取和保留日誌

日誌是疑難排解問題的有用資源。在使用日誌對 AWS ParallelCluster 資源問題進行疑難排解之前，您應該先建立叢集日誌的封存。請遵循 AWS ParallelCluster GitHub Wiki 上建立叢集日誌封存主題中所述的步驟，以啟動此程序。

如果其中一個執行中的叢集遇到問題，您應該先執行 `pcluster stop` <*cluster_name*>命令，將叢集置於 STOPPED 狀態，然後再開始疑難排解。這可防止產生任何非預期的成本。

如果 `pcluster` 停止運作，或如果您仍保留其日誌時要刪除叢集，請執行 `pcluster delete --keep-logs` <*cluster_name*>命令。執行此命令會刪除叢集，但會保留存放在 Amazon 中的日誌群組 CloudWatch。如需此命令的詳細資訊，請參閱 `pcluster delete` 文件。

對堆疊部署問題進行故障診斷

如果您的叢集無法建立並復原堆疊建立，您可以查看下列日誌檔案來診斷問題。您想要在這些日誌ROLLBACK_IN_PROGRESS中尋找 的輸出。失敗訊息應如下所示：

```
$ pcluster create mycluster
Creating stack named: parallelcluster-mycluster
Status: parallelcluster-mycluster - ROLLBACK_IN_PROGRESS
Cluster creation failed.  Failed events:
  - AWS::EC2::Instance MasterServer Received FAILURE signal with UniqueId
 i-07af1cb218dd6a081
```

若要診斷問題，請使用 再次建立叢集`pcluster create`，包括 `--norollback`旗標。然後，SSH在叢集中：

```
$ pcluster create mycluster --norollback
...
$ pcluster ssh mycluster
```

登入主機節點後，您應該會找到三個主要日誌檔案，可用來找出錯誤。

- `/var/log/cfn-init.log` 是`cfn-init`指令碼的日誌。首先檢查此日誌。您可能會在此日誌`Command chef failed`中看到類似 的錯誤。如需與錯誤訊息連線的詳細資訊，請參閱此行前面的行。如需詳細資訊，請參閱 cfn-init 。
- `/var/log/cloud-init.log` 是 cloud-init 的日誌。如果您在 中沒有看到任何內容`cfn-init.log`，請嘗試在下一步檢查此日誌。
- `/var/log/cloud-init-output.log` 是 Cloud-init 執行的命令輸出。這包括來自 的輸出`cfn-init`。在大多數情況下，您不需要查看此日誌來疑難排解此類問題。

對多個佇列模式叢集的問題進行疑難排解

本節與使用 2.9.0 版及更新 AWS ParallelCluster 版本搭配 所安裝的叢集相關 Slurm 任務排程器。如需多個佇列模式的詳細資訊，請參閱 多重佇列模式。

主題

- 金鑰日誌
- 對節點初始化問題進行故障診斷
- 對非預期節點替換和終止進行故障診斷
- 更換、終止或關閉問題執行個體和節點的電源
- 對其他已知節點和任務問題進行故障診斷

金鑰日誌

下表提供主機節點金鑰日誌的概觀：

`/var/log/cfn-init.log`

這是 AWS CloudFormation 初始日誌。它包含設定執行個體時執行的所有命令。有助於對初始化問題進行疑難排解。

`/var/log/chef-client.log`

這是 Chef 用戶端日誌。它包含透過 Chef/ 執行的所有命令CINC。有助於對初始化問題進行疑難排解。

`/var/log/parallelcluster/slurm_resume.log`

這是`ResumeProgram`日誌。它啟動動態節點的執行個體，有助於疑難排解動態節點啟動問題。

`/var/log/parallelcluster/slurm_suspend.log`

這是`SuspendProgram`日誌。當動態節點的執行個體終止時稱為 ，有助於疑難排解動態節點終止問題。檢查此日誌時，您也應該檢查`clustermgtd`日誌。

`/var/log/parallelcluster/clustermgtd`

這是`clustermgtd`日誌。它作為集中常駐程式執行，管理大多數叢集操作動作。它有助於對任何啟動、終止或叢集操作問題進行疑難排解。

`/var/log/slurmctld.log`

這是 Slurm control daemon log. AWS ParallelCluster doesn 不會做出擴展決策。相反地，它只會嘗試啟動資源來滿足 Slurm 要求。它適用於擴展和分配問題、任務相關問題，以及任何排程器相關的啟動和終止問題。

以下是運算節點的關鍵備註：

`/var/log/cloud-init-output.log`

這是 Cloud-init 日誌。它包含設定執行個體時執行的所有命令。有助於對初始化問題進行疑難排解。

`/var/log/parallelcluster/computemgtd`

這是`computemgtd`日誌。它會在每個運算節點上執行，以監控頭節點上`clustermgtd`常駐程式離線的罕見事件中的節點。有助於疑難排解非預期的終止問題。

/var/log/slurmd.log

這是 Slurm 運算常駐程式日誌。它有助於對初始化和運算失敗相關問題進行疑難排解。

對節點初始化問題進行故障診斷

本節說明如何對節點初始化問題進行疑難排解。這包括節點無法啟動、開啟電源或加入叢集的問題。

主節點：

適用的日誌：

- /var/log/cfn-init.log
- /var/log/chef-client.log
- /var/log/parallelcluster/clustermgtd
- /var/log/parallelcluster/slurm_resume.log
- /var/log/slurmctld.log

檢查 /var/log/cfn-init.log和 /var/log/chef-client.log 日誌。這些日誌應包含設定主機節點時執行的所有動作。在設定期間發生的大多數錯誤都應在/var/log/chef-client.log日誌中顯示錯誤訊息。如果在叢集的組態中指定預先安裝或安裝後指令碼，請再次檢查指令碼是否透過日誌訊息成功執行。

建立叢集時，主機節點需要等待運算節點加入叢集，才能加入叢集。因此，如果運算節點無法加入叢集，則主機節點也會失敗。您可以根據使用的運算備註類型，遵循以下一組程序之一，以對此類問題進行疑難排解：

動態運算節點：

- 搜尋運算節點名稱的ResumeProgram日誌（/var/log/parallelcluster/slurm_resume.log），以查看是否ResumeProgram曾經使用節點呼叫。（如果ResumeProgram從未呼叫，您可以檢查slurmctld日誌（/var/log/slurmctld.log）以確定是否 Slurm 曾經嘗試ResumeProgram使用節點呼叫。）
- 請注意，不正確的 許可ResumeProgram可能會導致 靜音ResumeProgram失敗。如果您使用自訂AMI搭配修改進行ResumeProgram設定，請檢查 ResumeProgram 是否為slurm使用者所有，並具有 744（rwxr--r--）許可。
- 如果 ResumeProgram 已呼叫，請檢查是否已為節點啟動執行個體。如果未啟動任何執行個體，您應該能夠看到描述啟動失敗的錯誤訊息。

- 如果執行個體已啟動，則設定程序期間可能會發生問題。您應該會從ResumeProgram日誌中看到對應的私有 IP 地址和執行個體 ID。此外，您可以查看特定執行個體的對應設定日誌。如需使用運算節點疑難排解設定錯誤的詳細資訊，請參閱下一節。

靜態運算節點：

- 檢查 clustermgtd（/var/log/parallelcluster/clustermgtd）日誌，查看是否啟動節點的執行個體。如果未啟動，應該會出現明確錯誤訊息，詳細說明啟動失敗。
- 如果啟動執行個體，則設定過程中會出現一些問題。您應該會從ResumeProgram日誌中看到對應的私有 IP 地址和執行個體 ID。此外，您可以查看特定執行個體的對應設定日誌。

- 運算節點：
  - 適用的日誌：
    - /var/log/cloud-init-output.log
    - /var/log/slurmd.log
  - 如果啟動運算節點，請先檢查 /var/log/cloud-init-output.log，其中應包含類似於主機節點上日誌的設定/var/log/chef-client.log日誌。在設定期間發生的大多數錯誤都應在/var/log/cloud-init-output.log日誌中顯示錯誤訊息。如果在叢集組態中指定預先安裝或安裝後指令碼，請檢查它們是否已成功執行。
  - 如果您使用AMI修改為 的自訂 Slurm 組態，則可能會有 Slurm 相關錯誤，導致運算節點無法加入叢集。如需排程器相關錯誤，請檢查/var/log/slurmd.log日誌。

對非預期節點替換和終止進行故障診斷

本節繼續探索如何疑難排解節點相關問題，特別是當節點意外更換或終止時。

- 適用的日誌：
  - /var/log/parallelcluster/clustermgtd（主節點）
  - /var/log/slurmctld.log（主節點）
  - /var/log/parallelcluster/computemgtd（運算節點）
- 節點意外更換或終止
  - 檢查clustermgtd日誌（/var/log/parallelcluster/clustermgtd）中的 ，查看 是否clustermgtd採取 動作來取代或終止節點。請注意， clustermgtd會處理所有一般節點維護動作。

  - 如果clustermgtd已取代或終止節點，則應該會出現一則訊息，詳細說明為什麼在節點上採取此動作。如果原因與排程器相關（例如，因為節點位於 中DOWN），請在 slurmctld 日誌中查看詳細資訊。如果原因與 Amazon EC2相關，應該會有資訊性訊息詳細說明需要替換的 Amazon EC2 相關問題。
  - 如果 clustermgtd未終止節點，請先檢查這是否為 Amazon 的預期終止EC2，更具體而言是位置終止。如果 clustermgtd 確定為運作狀態不佳，computemgtd則在運算節點上執行的也可以採取 動作來終止節點。檢查computemgtd日誌（/var/log/parallelcluster/computemgtd）以查看節點是否computemgtd終止。
- 節點失敗
  - 簽入slurmctld日誌（/var/log/slurmctld.log）以查看任務或節點失敗的原因。請注意，如果節點失敗，任務會自動重新排入佇列。
  - 如果slurm_resume報告啟動節點，並在幾分鐘後clustermgtd報告EC2該節點的 Amazon 中沒有對應的執行個體，則節點可能會在設定期間失敗。若要從運算（/var/log/cloud-init-output.log）擷取日誌，請執行下列步驟：
    - 提交任務以允許 Slurm 啟動新的節點。
    - 節點啟動後，使用此命令啟用終止保護。

      ```
      aws ec2 modify-instance-attribute --instance-id i-xyz --disable-api-termination
      ```

    - 使用此命令從節點擷取主控台輸出。

      ```
      aws ec2 get-console-output --instance-id i-xyz --output text
      ```

更換、終止或關閉問題執行個體和節點的電源

- 適用的日誌：
  - /var/log/parallelcluster/clustermgtd（主節點）
  - /var/log/parallelcluster/slurm_suspend.log（主節點）
- 在大多數情況下， 會clustermgtd處理所有預期的執行個體終止動作。請檢查clustermgtd日誌，了解為何無法取代或終止節點。
- 對於失敗的動態節點scaledown_idletime，請檢查SuspendProgram日誌，查看 是否SuspendProgram以特定節點作為引數slurmctld呼叫 。請注意， 實際上SuspendProgram不會執行任何動作。相反地，它只會在呼叫時記錄。所有執行個體終止和NodeAddr重設都由 完成clustermgtd。Slurm SuspendTimeout 會自動將節點置於 之後POWER_SAVING的狀態。

對其他已知節點和任務問題進行故障診斷

另一種已知問題類型是 AWS ParallelCluster 可能無法配置任務或做出擴展決策。對於此類問題， AWS ParallelCluster 僅根據 啟動、終止或維護資源 Slurm 指示。針對這些問題，請檢查`slurmctld`日誌以疑難排解這些問題。

對單一佇列模式叢集的問題進行疑難排解

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

本節適用於沒有多個佇列模式的叢集，具有下列兩種組態之一：

- 使用 2.9.0 及 之前的 AWS ParallelCluster 版本啟動 SGE, Torque、 或 Slurm 任務排程器。
- 使用 2.9.0 AWS ParallelCluster 版或更新版本啟動，以及 SGE 或 Torque 任務排程器。

主題

金鑰日誌

下列日誌檔案是主機節點的金鑰日誌。

對於 AWS ParallelCluster 2.9.0 版或更新版本：

`/var/log/chef-client.log`

這是 CINC（chef） 用戶端日誌。它包含透過 執行的所有命令CINC。它有助於對初始化問題進行疑難排解。

對於所有 AWS ParallelCluster 版本：

`/var/log/cfn-init.log`

這是`cfn-init`日誌。它包含設定執行個體時執行的所有命令，因此有助於對初始化問題進行疑難排解。如需詳細資訊，請參閱 cfn-init 。

`/var/log/clustermgtd.log`

這是 的`clustermgtd`日誌 Slurm 排程器。 會作為集中常駐程式`clustermgtd`執行，以管理大多數叢集操作動作。它有助於對任何啟動、終止或叢集操作問題進行疑難排解。

`/var/log/jobwatcher`

這是 的`jobwatcher`日誌 SGE 以及 Torque 排程器。 會`jobwatcher`監控排程器佇列並更新 Auto Scaling 群組。它有助於疑難排解與擴展節點相關的問題。

`/var/log/sqswatcher`

這是 的`sqswatcher`日誌 SGE 以及 Torque 排程器。 會在成功初始化後`sqswatcher`處理運算執行個體傳送的執行個體就緒事件。它也會將運算節點新增至排程器組態。此日誌有助於疑難排解節點或節點無法加入叢集的原因。

以下是運算節點的金鑰日誌。

AWS ParallelCluster 2.9.0 版或更新版本

`/var/log/cloud-init-output.log`

這是 Cloud init 日誌。它包含設定執行個體時執行的所有命令。它有助於對初始化問題進行疑難排解。

AWS ParallelCluster 2.9.0 之前的版本

`/var/log/cfn-init.log`

這是 CloudFormation 初始日誌。它包含設定執行個體時執行的所有命令。它有助於對初始化問題進行疑難排解

所有版本

/var/log/nodewatcher

這是使用 nodewatcher 時在每個運算節點上執行的 nodewatcher log. 精靈 SGE 以及 Torque 排程器。如果節點處於閒置狀態，它們會縮減節點。此日誌對於任何與縮減資源相關的問題很有用。

故障診斷失敗的啟動和聯結操作

- 適用的日誌：
  - /var/log/cfn-init-cmd.log（主節點和運算節點）
  - /var/log/sqswatcher（主節點）
- 如果節點無法啟動，請在/var/log/cfn-init-cmd.log日誌中查看特定錯誤訊息。在大多數情況下，節點啟動失敗是由於設定失敗所致。
- 如果運算節點在設定成功後仍無法加入排程器組態，請檢查/var/log/sqswatcher日誌，查看是否sqswatcher處理事件。在大多數情況下，這些問題都是因為 sqswatcher 未處理事件。

對擴展問題進行故障診斷

- 適用的日誌：
  - /var/log/jobwatcher（主節點）
  - /var/log/nodewatcher（運算節點）
- 擴展問題：針對主節點，請檢查/var/log/jobwatcher日誌，查看jobwatcher常駐程式是否計算了適當數量的必要節點，並更新了 Auto Scaling 群組。請注意， 會jobwatcher監控排程器佇列並更新 Auto Scaling 群組。
- 縮減問題規模：對於運算節點，請檢查問題節點上的/var/log/nodewatcher日誌，以了解節點縮減的原因。請注意，nodewatcher如果常駐程式處於閒置狀態，則會縮減運算節點。

其他叢集相關問題的疑難排解

一個已知問題是大規模叢集上的隨機運算備註失敗，特別是具有 500 個或更多運算節點的叢集。此問題與單一佇列叢集擴展架構的限制有關。如果您想要使用大規模叢集，使用 AWS ParallelCluster 2.9.0 版或更新版本，則使用 Slurm，並且想要避免此問題，您應該升級並切換到支援多個佇列模式的叢集。您可以執行 來執行此操作[pcluster-config convert](#)。

對於超大型叢集，可能需要對系統進行額外的調校。如需詳細資訊，請聯絡 AWS Support。

置放群組和執行個體啟動問題

若要取得最低節點間延遲，請使用置放群組。置放群組可確保您的執行個體位於相同的網路骨幹上。如果提出請求時沒有足夠的執行個體可用，則會傳回InsufficientInstanceCapacity錯誤。若要降低在使用叢集置放群組時收到此錯誤的可能性，請將 placement_group 參數設定為 ，DYNAMIC並將 placement 參數設定為 compute。

如果您需要高效能共用檔案系統，請考慮使用 FSx for Lustre。

如果主機節點必須位於置放群組中，請同時針對主機和所有運算節點使用相同的執行個體類型和子網路。透過執行此操作， compute_instance_type 參數具有與 master_instance_type 參數相同的值， placement 參數設定為 cluster，並且compute_subnet_id參數未指定。使用此組態時，master_subnet_id 參數的值會用於運算節點。

如需詳細資訊，請參閱 Amazon EC2使用者指南中的對執行個體啟動問題和置放群組角色和限制進行故障診斷 https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#concepts-placement-groups

無法取代的目錄

下列目錄會在節點之間共用，且無法取代。

/home

這包括預設的使用者主資料夾（ /home/ec2_user在 Amazon Linux 上，在 /home/centos上 CentOS、 和 /home/ubuntu 於 Ubuntu).

/opt/intel

這包括 Intel MPI、Intel Parallel Studio 和相關檔案。

/opt/sge

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

這包括 Son of Grid Engine 和相關檔案。(有條件，僅在 scheduler = sge 的時候。)

/opt/slurm

這包括 Slurm Workload Manager 和相關檔案。(有條件，僅在 scheduler = slurm 的時候。)

`/opt/torque`

> **Note**
>
> 從 2.11.5 版開始， AWS ParallelCluster 不支援使用 SGE 或 Torque 排程器。

這包括 Torque Resource Manager 和相關檔案。(有條件，僅在 `scheduler = torque` 的時候。)

對 Amazon 中的問題進行故障診斷 DCV

主題

Amazon 的日誌 DCV

Amazon 的日誌DCV會寫入 `/var/log/dcv/`目錄中的檔案。檢閱這些日誌有助於對問題進行疑難排解。

Amazon DCV執行個體類型記憶體

執行個體類型至少要有 1.7 GB （ GiB ） RAM才能執行 Amazon DCV。Nano 以及 micro 執行個體類型沒有足夠的記憶體來執行 Amazon DCV。

Ubuntu Amazon DCV問題

透過 Ubuntu 上的DCV工作階段執行 Gnome 終端機時，您可能無法自動存取透過登入 Shell AWS ParallelCluster 提供的使用者環境。使用者環境提供 Openmpi 或 intelmpi 等環境模組，以及其他使用者設定。

Gnome Terminal 的預設設定可防止 Shell 作為登入 Shell 啟動。這表示系統不會自動取得 Shell 設定檔，也不會載入 AWS ParallelCluster 使用者環境。

若要正確取得 Shell 設定檔並存取 AWS ParallelCluster 使用者環境，請執行下列其中一項操作：

- 變更預設終端機設定：
  1. 選擇 Gnome 終端機中的編輯選單。
  2. 選取偏好設定 ，然後選取設定檔 。
  3. 選擇 命令，然後選擇 執行命令 作為登入 shell 。
  4. 開啟新的終端機。
- 使用命令列來尋找可用的設定檔：

```
$ source /etc/profile && source $HOME/.bashrc
```

透過整合對叢集 AWS Batch 中的問題進行故障診斷

本節與具有 AWS Batch 排程器整合的叢集相關。

頭節點問題

與主機節點相關的設定問題可以與單一佇列叢集相同的方式進行故障診斷。如需有關這些問題的詳細資訊，請參閱對單一佇列模式叢集的問題進行疑難排解。

AWS Batch 多節點平行任務提交問題

如果您在使用 AWS Batch 作為任務排程器時提交多節點平行任務時遇到問題，您應該升級至 2.5.0 AWS ParallelCluster 版。如果不可行，您可以使用主題中詳述的解決方法：自行修補用於透過 提交多節點平行任務的叢集 AWS Batch。

運算問題

AWS Batch 管理服務的擴展和運算層面。如果您遇到運算相關問題，請參閱故障 AWS Batch 診斷文件以取得協助。

任務失敗

如果任務失敗，您可以執行 `awsbout`命令來擷取任務輸出。您也可以執行 `awsbstat -d`命令，以取得 Amazon 所儲存之任務日誌的連結 CloudWatch。

資源無法建立時的故障診斷

本節與叢集資源在無法建立時有關。

當資源無法建立時，會 ParallelCluster 傳回如下錯誤訊息。

```
pcluster create -c config my-cluster
Beginning cluster creation for cluster: my-cluster
WARNING: The instance type 'p4d.24xlarge' cannot take public IPs. Please make sure that
 the subnet with
id 'subnet-1234567890abcdef0' has the proper routing configuration to allow private IPs
 reaching the
Internet (e.g. a NAT Gateway and a valid route table).
WARNING: The instance type 'p4d.24xlarge' cannot take public IPs. Please make sure that
 the subnet with
id 'subnet-1234567890abcdef0' has the proper routing configuration to allow private IPs
 reaching the Internet
(e.g. a NAT Gateway and a valid route table).
Info: There is a newer version 3.0.3 of AWS ParallelCluster available.
Creating stack named: parallelcluster-my-cluster
Status: parallelcluster-my-cluster - ROLLBACK_IN_PROGRESS
Cluster creation failed.  Failed events:
- AWS::CloudFormation::Stack MasterServerSubstack Embedded stack
arn:aws:cloudformation:region-id:123456789012:stack/parallelcluster-my-cluster-
MasterServerSubstack-ABCDEFGHIJKL/a1234567-b321-c765-d432-dcba98766789
was not successfully created:
The following resource(s) failed to create: [MasterServer].
- AWS::CloudFormation::Stack parallelcluster-my-cluster-MasterServerSubstack-
ABCDEFGHIJKL The following resource(s) failed to create: [MasterServer].
- AWS::EC2::Instance MasterServer You have requested more vCPU capacity than your
 current vCPU limit of 0 allows for the instance bucket that the
specified instance type belongs to. Please visit http://aws.amazon.com/contact-us/ec2-
request to request an adjustment to this limit.
(Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request ID:
 a9876543-b321-c765-d432-dcba98766789; Proxy: null)
}
```

例如，如果您看到上一個命令回應中顯示的狀態訊息，則必須使用不超過目前 vCPU 限制的執行個體類型，或請求更多 v CPU容量。

您也可以使用 CloudFormation 主控台來查看 "Cluster creation failed" 狀態的相關資訊。

從主控台檢視 CloudFormation 錯誤訊息。

1. 登入 AWS Management Console 並導覽至 https://console.aws.amazon.com/cloudformation 。
2. 選取名為 parallelcluster- 的堆疊*cluster_name*.

3. 選擇事件索引標籤。
4. 透過按邏輯 ID 捲動資源事件清單，檢查無法建立的資源狀態。如果子任務無法建立，請向後工作以尋找失敗的資源事件。
5. AWS CloudFormation 錯誤訊息的範例：

```
2022-02-07 11:59:14 UTC-0800 MasterServerSubstack CREATE_FAILED Embedded stack
arn:aws:cloudformation:region-id:123456789012:stack/parallelcluster-my-cluster-
MasterServerSubstack-ABCDEFGHIJKL/a1234567-b321-c765-d432-dcba98766789
was not successfully created: The following resource(s) failed to create:
 [MasterServer].
```

對IAM政策大小問題進行故障診斷

請參閱 IAM 和 AWS STS 配額、名稱要求和字元限制，以驗證連接到角色的受管政策的配額。如果受管政策大小超過配額，請將政策分割為兩個或更多政策。如果您超過附加至IAM角色的政策數量配額，請建立其他角色並在這些角色之間分發政策以符合配額。

其他支援

如需已知問題的清單，請參閱 GitHub Wiki 主頁面或問題頁面。如有更緊急的問題，請聯絡 AWS Support 或開啟新 GitHub問題 。

AWS ParallelCluster支援政策

AWS ParallelCluster同時支援多個發行版本。每個AWS ParallelCluster版本都有排定的支援終止 (EOSL) 日期。在 EOSL 日期之後，不會針對該版本提供進一步的支援或維護。

AWS ParallelCluster使用`major.minor.`*`patch`*版本配置。新功能、效能改善、安全性更新和錯誤修正都包含在最新主要版本的新次要版本發行版本中。次要版本在主要版本中向後兼容。針對重大問題，可透過修補程式版本AWS提供修正程式，但僅適用於尚未達到 EOSL 的最新次要版本。如果您想要使用新版本的更新，則需要升級至新的次要版本或修補程式版本。

AWS ParallelCluster 版本	支援生命週期結束 (EOSL) 日期
2.10.4 及更早版本	2021 年 12 月 31 日
2.11. *x*	2022 年 12 月 31 日

AWS ParallelCluster 中的安全性

雲端安全是 AWS 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。

安全是 AWS 與您共同肩負的責任。共同責任模型將其描述為雲端本身的安全和雲端內部的安全：

- 雲端本身的安全 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。第三方稽核人員會定期測試和驗證我們安全性的有效性，作為 AWS 合規計劃的一部分。若要了解適用於 AWS ParallelCluster 的合規計劃，請參閱合規計劃的 AWS 服務範圍。
- 雲端安全性 — 您的責任取決於您使用的特AWS定服務或服務。您還需要對其他幾個相關因素負責，包括數據的敏感性，公司的要求以及適用的法律和法規。

本文件說明您在使用時應如何套用共同的責任模型AWS ParallelCluster。下列主題說明如何將 AWS ParallelCluster 設定為達到您的安全及合規目標。您還將學習如何以AWS ParallelCluster幫助您監控和保護AWS資源的方式使用。

主題

- 所使用服務的安全性資訊 AWS ParallelCluster
- 資料保護 AWS ParallelCluster
- AWS ParallelCluster 的 Identity and Access Management
- AWS ParallelCluster 的合規驗證
- 強制執行最低版本為 TLS 1.2

所使用服務的安全性資訊 AWS ParallelCluster

- 亞馬遜 EC2 中的安全性
- 亞馬遜 API 閘道中的安全性
- 中的安全性 AWS Batch
- 中的安全性 AWS CloudFormation
- 亞馬遜的安全性 CloudWatch
- 中的安全性 AWS CodeBuild
- 亞馬遜動態 B 中的安全性

- 亞馬遜 ECR 中的安全性
- 亞馬遜 ECS 中的安全性
- 亞馬遜 EFS 的安全性
- FSx 中針對光澤的安全性
- AWS Identity and Access Management(IAM) 中的安全性
- EC2 映像產生器中的安全性
- 中的安全性 AWS Lambda
- 亞馬遜路線 53 中的安全性
- 亞馬遜 SNS 中的安全性
- 亞馬遜 SQS 中的安全性（適用於 2.x AWS ParallelCluster 版）
- 亞馬遜 S3 中的安全性
- 亞馬遜 VPC 中的安全性

資料保護 AWS ParallelCluster

所以此 AWS 共同責任模型適用於資料保護 AWS ParallelCluster。 如本模型所述， AWS 負責保護運行所有的全球基礎設施 AWS 雲端。 您有責任維持對託管在此基礎結構上的內容的控制權。您也必須負責 AWS 服務 你使用的。如需有關資料隱私權的詳細資訊，請參閱資料隱私權FAQ。 如需歐洲資料保護的相關資訊，請參閱 AWS 共同責任模型和GDPR博客文章 AWS 安全部落格。

出於數據保護目的，我們建議您進行保護 AWS 帳戶 憑據並設置個別用戶 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM)。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：

- 對每個帳戶使用多重要素驗證 (MFA)。
- 使用SSL/TLS與之溝通 AWS 的費用。我們需要 TLS 1.2 並推薦 TLS 1.3。
- 設定API和使用者活動記錄 AWS CloudTrail。 如需使用 CloudTrail 軌跡進行擷取的相關資訊 AWS 活動，請參閱使用 CloudTrail 系統線 AWS CloudTrail 使用者指南。
- 使用 AWS 加密解決方案，以及其中的所有默認安全控制 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。
- 如果您在訪問時需要 FIPS 140-3 驗證的加密模塊 AWS 透過指令行介面或API使用FIPS端點。如需有關可用FIPS端點的詳細資訊，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 ParallelCluster 或其他 AWS 服務 使用控制台API， AWS CLI，或 AWS SDKs。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URL給外部伺服器，我們強烈建議您不要在中包含認證資訊，URL以驗證您對該伺服器的要求。

資料加密

任何安全服務都有一項重要功能，就是當資訊處於非使用中狀態時，就會將資訊加密。

靜態加密

AWS ParallelCluster 本身不會儲存任何客戶資料，而非與之互動所需的憑證 AWS 代表使用者提供的服務。

對於叢集中節點上的資料，可以在靜態時加密資料。

對於 Amazon EBS 磁碟區，加密是使用下列區[ebs]段中的ebs_kms_key_id設定來設定 AWS ParallelCluster 版本 2.x。）如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Amazon EBS 加密。

對於 Amazon EFS 磁碟區，加密是使用中一[efs]節中的encrypted和efs_kms_key_id設定進行設定 AWS ParallelCluster 版本 2.x）。如需詳細資訊，請參閱 Amazon Elastic File System 使用者指南中的靜態加密運作方式。

對FSx於 Lustre 檔案系統，在建立 Amazon FSx 檔案系統時，會自動啟用靜態資料加密。如需詳細資訊，請參閱 Amazon for Lustre 使FSx用者指南中的加密靜態資料。

對於具有NVMe磁碟區的執行個體類型，NVMe執行個體儲存磁碟區上的資料會使用在執行個體的硬體模組上實作 XTS AES -256 密碼來加密。加密金鑰是使用硬體模組產生的，每個NVMe執行個體儲存裝置都是唯一的。所有加密金鑰會在執行個體停止或終止時銷毀，且無法復原。您無法停用此加密，也無法提供您自己的加密金鑰。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的靜態加密。

如果您使用 AWS ParallelCluster 調用 AWS 該服務會將客戶資料傳輸至您的本機電腦以進行儲存，然後參閱該服務使用指南中的「安全性與合規性」一章，以取得有關如何儲存、保護和加密資料的資訊。

傳輸中加密

根據預設，從用戶端電腦傳輸的所有資料都在執行 AWS ParallelCluster 以及 AWS 通過HTTPS/TLS連接發送所有內容來加密服務端點。視選取的執行個體類型而定，叢集中節點之間的流量可以自動加密。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的傳輸中加密。

另請參閱

- Amazon 的數據保護 EC2
- EC2Image Builder 中的資料保護
- 資料保護 AWS CloudFormation
- Amazon 的數據保護 EFS
- Amazon S3 中的資料保護
- 資料保護中FSx的光澤

AWS ParallelCluster 的 Identity and Access Management

AWS ParallelCluster使用角色來存取您的AWS資源及其服務。AWS ParallelCluster用來授與權限的執行個體和使用者原則記錄在AWS Identity and Access Management 中的 角色 AWS ParallelCluster。

唯一的主要差異在於使用標準 使用者和長期登入資料時的驗證方式。雖然使用者需要密碼才能存取AWS服務的主控台，但是該使用者需要使用存取金鑰組才能執行相同的作業AWS ParallelCluster。所有其他短期憑證的使用方式都與搭配主控台使用的方式一樣。

所使用的認證會儲存AWS ParallelCluster在純文字檔案中，而且不會加密。

- `$HOME/.aws/credentials` 檔案會存放存取 AWS 資源所需的長期憑證。這些包含您的存取金鑰 ID 和私密存取金鑰。
- 短期憑證，像是您擔任的角色，或用於 AWS IAM Identity Center 服務的角色，也會分別儲存於 `$HOME/.aws/cli/cache` 和 $HOME/.aws/sso/cache 資料夾中。

降低風險

- 強烈建議您設定 `$HOME/.aws` 資料夾及其子資料夾和檔案的檔案系統權限，以限制為僅供授權使用者進行存取。
- 盡可能使用具有臨時憑證的角色，以在憑證洩漏時減少損害的機會。僅將長期憑證用於請求及重新整理短期角色憑證。

AWS ParallelCluster 的合規驗證

在多個 AWS 合規計畫中，第三方稽核人員會評估 AWS 服務的安全與合規。使AWS ParallelCluster用存取服務並不會改變該服務的合規性。

如需特定合規計劃的 AWS 服務範圍清單，請參閱合規計劃的 AWS 服務範圍。如需一般資訊，請參閱 AWS 合規計劃。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱下載 AWS Artifact 中的報告。

您使用 AWS ParallelCluster 時的合規責任取決於資料的敏感度、您的公司的合規目標，以及適用的法律和法規。AWS 提供以下資源協助您處理合規事宜：

- 安全性與合規快速入門指南 – 這些部署指南會描述架構考量，並提供在 AWS 上部署以安全性及合規為重心之基準環境的步驟。
- 在 Amazon 網路服務上架構 HIPAA 安全性與合規性白皮書 — 本AWS白皮書說明公司如何使用建立符合 HIPAA 規範的應用AWS程式。
- AWS 合規資源 – 這組手冊和指南可能適用於您的產業和所處位置。
- 《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在內部實務、業界準則和法規方面的合規程度。
- AWS Security Hub – 此 AWS 服務可供您檢視 AWS 中的安全狀態，可助您檢查是否符合安全產業標準和最佳實務。

強制執行最低版本為 TLS 1.2

若要在與AWS服務通訊時增加安全性，您應該將您的設定AWS ParallelCluster為使用 TLS 1.2 或更新版本。當您使用時AWS ParallelCluster，Python 用於設置 TLS 版本。

為了確保不AWS ParallelCluster使用 TLS 1.2 之前的 TLS 版本，您可能需要重新編譯 OpenSSL 以強制執行此最低限度，然後重新編譯 Python 以使用新構建的 OpenSSL。

判定目前支援的通訊協定

首先，使用 OpenSSL 建立用於測試伺服器和 Python SDK 的自簽憑證。

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out
 cert.pem -days 365
```

然後使用 OpenSSL 啟動測試伺服器。

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

在新的終端機視窗中，建立虛擬環境並安裝 Python SDK。

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

建立一個名為 check.py 的 Python 指令碼，此指令碼使用 SDK 的基礎 HTTP 程式庫。

```
$ import urllib3
URL = 'https://localhost:4433/'

http = urllib3.PoolManager(
ca_certs='cert.pem',
cert_reqs='CERT_REQUIRED',
)
r = http.request('GET', URL)
print(r.data.decode('utf-8'))
```

執行新的指令碼。

```
$ python check.py
```

這會顯示有關所建立連線的詳細資訊。在輸出中搜尋 "Protocol : " (通訊協定：)。如果輸出是 "TLSv1.2" 或更新版本，SDK 就預設為 TLS v1.2 或更新版本。如果是較早的版本，您就必須重新編譯 OpenSSL 再重新編譯 Python。

但是，即使您安裝的 Python 預設為 TLS v1.2 或更新版本，如果伺服器不支援 TLS v1.2 或更新版本，則 Python 仍然可能必須與 TLS v1.2 更早的版本重新交涉。若要確保 Python 不會自動與較早版本重新交涉，請使用以下命令重新啟動測試伺服器。

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

如果您使用的是較早版本的 OpenSSL，您可能無法使用 -no_tls_3 旗標。如果是這種情況，請刪除該旗標，因為您使用的 OpenSSL 版本不支援 TLS v1.3。然後執行 Python 指令碼。

```
$ python check.py
```

如果您的 Python 安裝正確，不會與 TLS 1.2 之前的版本重新交涉，您應該會收到 SSL 錯誤。

```
$ urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='localhost',
 port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL:
 UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

如果能夠建立連線，則您必須重新編譯 OpenSSL 和 Python，以禁止與 TLS v1.2 之前的通訊協定交涉。

編譯 OpenSSL 和 Python

為了確保AWS ParallelCluster不會針對 TLS 1.2 之前的任何內容進行協商，您需要重新編譯 OpenSSL 和 Python。若要執行此操作，請複製下列內容以建立並執行此指令碼。

```
#!/usr/bin/env bash
set -e

OPENSSL_VERSION="1.1.1d"
OPENSSL_PREFIX="/opt/openssl-with-min-tls1_2"
PYTHON_VERSION="3.8.1"
PYTHON_PREFIX="/opt/python-with-min-tls1_2"


curl -O "https://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz"
tar -xzf "openssl-$OPENSSL_VERSION.tar.gz"
cd openssl-$OPENSSL_VERSION
./config --prefix=$OPENSSL_PREFIX no-ssl3 no-tls1 no-tls1_1 no-shared
make > /dev/null
sudo make install_sw > /dev/null


cd /tmp
curl -O "https://www.python.org/ftp/python/$PYTHON_VERSION/Python-$PYTHON_VERSION.tgz"
tar -xzf "Python-$PYTHON_VERSION.tgz"
cd Python-$PYTHON_VERSION
./configure --prefix=$PYTHON_PREFIX --with-openssl=$OPENSSL_PREFIX --disable-shared > /
dev/null
make > /dev/null
sudo make install > /dev/null
```

這樣會編譯一個 Python 版本，內含不會自動與 TLS 1.2 之前任何版本交涉的靜態連結 OpenSSL。這也會在 `/opt/openssl-with-min-tls1_2` 目錄中安裝 OpenSSL，並在 `/opt/python-with-min-tls1_2` 目錄中安裝 Python。執行此指令碼之後，確認已安裝新版本的 Python。

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

這應該會列印以下內容。

```
Python 3.8.1
```

若要確認這個新版本的 Python 不會與 TLS 1.2 之前的版本交涉，請使用新安裝的 Python 版本 (也就是 /opt/python-with-min-tls1_2/bin/python3) 重新執行 判定目前支援的通訊協定 的步驟。

版本備註和文件歷史記錄

下表說明 AWS ParallelCluster 使用者指南 的主要更新和新功能。我們也會經常更新文件，以處理您傳送給我們的意見回饋。

變更	描述	日期
僅文件版本	AWS ParallelCluster 已發佈第 2 版特定使用者指南。 僅文件版本： • AWS ParallelCluster 第 2 版有自己的個別使用者指南。	2023 年 7 月 17 日
AWS ParallelCluster 2.11.9 版已發行	AWS ParallelCluster 2.11.9 版已發行。 錯誤修正： • 防止取代 受管FSx的 Lustre 檔案系統，以及遺失包含 變更的叢集更新上的資料`vpc_security_group_id` 。 如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 套件`CHANGELOG` 檔案 GitHub。	2022 年 12 月 2 日
AWS ParallelCluster 2.11.8 版已發行	AWS ParallelCluster 2.11.8 版已發行。	2022 年 11 月 14 日

變更：

- 將 Intel MPI Library 升級到 2021 版更新 6（更新自 2021 版更新 4）。如需詳細資訊，請參閱 Intel® MPI Library 2021 Update 6。
- 將EFA安裝程式升級至 1.19.0
  - Efa-driver： `efa-1.16.0-1`
  - Efa-config： `efa-config-1.11-1` （來自 `efa-config-1.9-1` ）
  - Efa-profile：`efa-profile-1.5-1` （無變更）
  - Libfabric-aws： `libfabric-aws-1.16.0-1` （來自 `libfabric-1.13.2` ）
  - Rdma 核心： `rdma-core-41.0-2` （來自 `rdma-core-37.0` ）
  - 開啟 MPI： `openmpi40-aws-4.1.4-3` （從 `openmpi40-aws-4.1.1-2` ）
- 將 Lambda 函數在 AWS Batch 整合中使用的 Python 執行期升級至 python3.9。

錯誤修正：

- 防止叢集標籤在更新期間變更，因為不支援。

	如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 套件CHANGELOG 檔案 GitHub。	
AWS ParallelCluster 2.11.7 版已發行	AWS ParallelCluster 2.11.7 版已發行。 變更： • 升級 Slurm 至 20.11.9 版。 如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 套件CHANGELOG 檔案 GitHub。	2022 年 5 月 13 日
AWS ParallelCluster 2.11.6 版已發行	AWS ParallelCluster 2.11.6 版已發行。 增強功能： • 改善遺失聯網時的例外狀況管理。 變更： • 作業系統套件更新和安全性修正。 如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 套件CHANGELOG 檔案 GitHub。	2022 年 4 月 19 日

AWS ParallelCluster 2.11.5 版已發行	AWS ParallelCluster 2.11.5 版已發行。	2022 年 3 月 1 日

增強功能：

- 將 的支援新增NEW_CHANGED_DELETED 為FSx適用於 Lustre AutoImportPolicy 選項的值 。
- 移除 SGE和 扭矩排程器的支援。
- 在 Amazon Linux 上停用log4j-cve-2021-44228-hotpatch 服務，以避免造成潛在的效能降低。

變更：

- 將NVIDIA驅動程式升級至版本 470.103.01 （從 470.82.01 ）。
- 將 NVIDIA Fabric Manager 升級到版本 470.103.01 （從 470.82.01 ）。
- 將CUDA程式庫升級至版本 11.4.4（從 11.4.3）。
- Intel MPI 已更新至 2021 版更新 4（更新自 2019 版更新 8）。如需詳細資訊，請參閱 Intel® MPI Library 2021 Update 4。
- 將主機節點建立逾時延長為一小時。

錯誤修正：

- 透過瀏覽器修正DCV連線。
- 修正引YAML號，以防止自訂標籤剖析為數字。

如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 套件CHANGELOG 檔案 GitHub。

AWS ParallelCluster 2.11.4 版已發行	AWS ParallelCluster 2.11.4 版已發行。 變更包括： • CentOS 已移除 8 個支援。CentOS 8 會在 2021 年 12 月 31 日達到生命週期結束（EOL）。 • 升級 Slurm Workload Manager 至 20.11.8 版。 • 將 Cinc 用戶端升級至 17.2.29。 • Amazon DCV 已更新至 Amazon DCV 2021.2-11190。如需詳細資訊，請參閱 Amazon DCV管理員指南中的 DCV 2021.2-11190 — 2021 年 10 月 11 日。 • 將NVIDIA驅動程式升級至版本 470.82.01 （從 460.73.01 ）。 • 將CUDA程式庫升級至版本 11.4.3（從 11.3.0）。 • 將 NVIDIA Fabric Manager 升級到 470.82.01 。 • 在 Amazon Linux 2 上的執行個體啟動時間停用套件更新。 • 在 上停用全自動套件更新 Ubuntu 和 Amazon Linux 2。 • 在 上安裝 Python 3 AWS CloudFormation 版本的協助程式指令碼 CentOS 7 和	2021 年 12 月 20 日

Ubuntu 18.04.（這些已用於 Amazon Linux 2 和 Ubuntu 20.04.）

修正項目包括：

- 停用 `ec2_iam_role` 參數更新。
- 修正 啟動範本中的`CpuOptions` 組態 T2 執行個體。

如需變更的詳細資訊，請參閱 aws-parallelcluster `CHANGELOG` 的檔案，aws-parallelcluster-cookbook以及 上的aws-parallelcluster-node套件 GitHub。

AWS ParallelCluster 2.11.3 版已發行	AWS ParallelCluster 2.11.3 版已發行。 • 修正pcluster createami 失敗原因 Son of Grid Engine 來源無法在上使用arc.liv.ac.uk 。 將Elastic Fabric Adapter安裝程式升級至 1.14.1（從 1.13.0 版開始） • EFA 組態： efa-config-1.9-1 （來自 efa-config-1.9 ） • EFA 設定檔：efa-profile-1.5-1 （無變更） • EFA 核心模組：efa-1.14.2 （來自 efa-1.13.0 ） • RDMA 核心： rdma-core-37.0 （來自 rdma-core-35.0amzn ） • Libfabric：libfabric-1.13.2 （來自 libfabric-1.13.0amzn1.0 ） • 開啟 MPI： openmpi40-aws-4.1.1-2 （無變更） GPUDirect RDMA 如果執行個體類型支援，一律會啟用。	2021 年 11 月 3 日

- `enable_efa_gdr` 和 `enable_efa_gdr` 組態選項沒有效果。

如需變更的詳細資訊，請參閱 aws-parallelcluster CHANGELOG 的檔案，aws-parallelcluster-cookbook以及 上的aws-parallelcluster-node套件 GitHub。

AWS ParallelCluster 2.11.2 版已發行	AWS ParallelCluster 2.11.2 版已發行。
	
	變更包括：
	
	• 如果 安裝在 基礎 中EFA，請勿在開機時間啟用 EFA GPUDirectRDMA（GDR）的情況下安裝 AMI。
	• 鎖定nvidia-fabricmanager 套件版本，以保持與 所安裝的 NVIDIA驅動程式版本同步 AWS ParallelCluster。
	• Slurm：修正節點開機時，叢集停止和重新啟動時造成的問題。
	• Elastic Fabric Adapter 安裝程式已更新至 1.13.0：
	  • EFA 組態：efa-config-1.9 （無變更）
	  • EFA 設定檔：efa-profile-1.5-1 （無變更）
	  • EFA 核心模組：efa-1.13.0 （無變更）
	  • RDMA 核心： rdma-core-35.0amzn （來自 rdma-core-32.1amzn ）
	  • Libfabric：libfabric-1.13.0amzn1.0 （來自 libfabric-1.11.2amzn1.1 ）
		2021 年 8 月 27 日

- 開啟 MPI： `openmpi40-aws-4.1.1-2` （無變更）
- 將自訂與預先安裝的EFA套件AMI搭配使用時，EFA節點引導時間不會對 進行任何變更。保留原始EFA套件部署。

如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 和aws-parallelcluster-cookbook套件CHANGELOG檔案 GitHub。

AWS ParallelCluster 2.11.1 版已發行	AWS ParallelCluster 2.11.1 版已發行。
	
	變更包括：
	
	• 使用掛載選項noatime掛載檔案系統，以在讀取檔案時停止記錄上次存取時間。這可改善遠端檔案系統的效能。
	• Elastic Fabric Adapter 安裝程式已更新至 1.12.3：
	  • EFA 組態： efa-config-1.9 （來自 efa-config-1.8-1 ）
	  • EFA 設定檔：efa-profile-1.5-1 （無變更）
	  • EFA 核心模組： efa-1.13.0 （來自 efa-1.12.3 ）
	  • RDMA 核心：rdma-core-32.1amzn （無變更）
	  • Libfabric：libfabric-1.11.2amzn1.1 (未變更)
	  • 開啟 MPI： openmpi40-aws-4.1.1-2 （無變更）
	• 使用 AWS Batch 作為排程器時，在主機節點上重試安裝aws-parallelcluster 套件。
	• 建置時避免失敗 SGE 在執行個體類型上，超過 31 個 vCPUs。
		2021 年 7 月 23 日

- 釘選至 Amazon CloudWatch Agent 的 1.247347.6 版，以避免版本 1.247348.0 中出現的問題。

如需變更的詳細資訊，請參閱 上的 aws-parallelcluster 和aws-parallelcluster-cookbook套件CHANGELOG檔案 GitHub。

AWS ParallelCluster 2.11.0 版已發行	AWS ParallelCluster 2.11.0 版已發行。
	
	變更包括：
	
	• 新增 的支援 Ubuntu 20.04（ubuntu2004 ）和已移除對 的支援 Ubuntu 16.04（ubuntu1604 ）和 Amazon Linux（alinux）。Amazon Linux 2（alinux2）仍然完全支援。如需詳細資訊，請參閱base_os。
	• 已移除對 Python 3.6 以下版本的支援。
	• 預設根磁碟區大小增加至 35 GB（GiB ）。如需詳細資訊，請參閱 compute_root_volume_size 和 master_root_volume_size 。
	• Elastic Fabric Adapter 安裝程式已更新至 1.12.2：
	  • EFA 組態：efa-config-1.8-1 （來自 efa-config-1.7 ）
	  • EFA 設定檔：efa-profile-1.5-1 （來自 efa-profile-1.4 ）
	  • EFA 核心模組：efa-1.12.3 （來自 efa-1.10.2 ）
	  • RDMA 核心：rdma-core-32.1amzn
		2021 年 7 月 1 日

（來自 rdma-core -31.2amzn ）

- Libfabric： libfabric -1.11.2amzn1.1 （來自 libfabric -1.11.1amzn1.0 ）
- 開啟 MPI： openmpi40 -aws-4.1.1-2 （從 openmpi40-aws-4.1. 0 ）

- 已升級 Slurm 至 版本 20.11.7（從 20.02.7）。
- 在 centos7和 上安裝SSM 代理程式centos8。（SSM 代理程式已預先安裝在 alinux2、 ubuntu180 4 和 中ubuntu2004 。）
- SGE：一律使用短名稱作為主機名稱篩選條件與 qstat。
- 使用執行個體中繼資料服務第 2 版（IMDSv2）而非執行個體中繼資料服務第 1 版（IMDSv1）來擷取執行個體中繼資料。如需詳細資訊，請參閱 Amazon EC2使用者指南 中的執行個體中繼資料和使用者資料。
- 將NVIDIA驅動程式升級至版本 460.73.01 （從 450.80.02 ）。
- 將CUDA程式庫升級至版本 11.3.0（從 11.0）。

- 將 NVIDIA Fabric Manager 升級到 nvidia-fabricmanager-460 。
- 將 AWS ParallelCluster Virtualenvs 中使用的 Python 升級至 3.7.10（從 3.6.13）。
- 將 Cinc 用戶端升級至 16.13.16。
- 升級 的第三方相依性aws-parallelcluster-cookbook：
  - apt-7.4.0 （來自 apt-7.3.0 ）。
  - iptables-8.0.0 （來自 iptables-7.1.0 ）。
  - line-4.0.1 （來自 line-2.9.0 ）。
  - openssh-2.9.1 （來自 openssh-2.8.1 ）。
  - pyenv-3.4.2 （來自 pyenv-3.1.1 ）。
  - selinux-3.1.1 （來自 selinux-2.1.1 ）。
  - ulimit-1.1.1 （來自 ulimit-1.0.0 ）。
  - yum-6.1.1 （來自 yum-5.1.0 ）。
  - yum-epel-4.1.2 （來自 yum-epel-3.3.0 ）。

如需變更的詳細資訊，請參閱 上的 aws-parallelcluste

	r、aws-parallelcluster-cookbook和aws-parallelcluster-node套件CHANGELOG的檔案 GitHub。	
AWS ParallelCluster 2.10.4 版已發行	AWS ParallelCluster 2.10.4 版已發行。 變更包括： • 已升級 Slurm 至 版本 20.02.7（從 20.02.4）。 如需變更的詳細資訊，請參閱上的 aws-parallelcluster 套件CHANGELOG檔案 GitHub。	2021 年 5 月 15 日

AWS ParallelCluster 2.10.3 版已發行	AWS ParallelCluster 2.10.3 版已發行。
	
	變更包括：
	
	• 新增 的支援 Ubuntu AWS 中國 和 中以 Arm 為基礎的 AWS Graviton 執行個體上的 18.04 和 Amazon Linux 2 AWS GovCloud (US) AWS 區域。
	• Elastic Fabric Adapter 安裝程式已更新至 1.11.2：
	  • EFA 組態：efa-config-1.7 （無變更）
	  • EFA 設定檔： efa-profile-1.4 （來自 efa-profile-1.3 ）
	  • EFA 核心模組： efa-1.10.2 （無變更）
	  • RDMA 核心：rdma-core-31.2amzn （無變更）
	  • Libfabric：libfabric-1.11.1amzn1.0 (未變更)
	  • 開啟 MPI： openmpi40-aws-4.1.0 （無變更）
	
	如需變更的詳細資訊，請參閱上的 aws-parallelcluster 套件 CHANGELOG檔案 GitHub。
		2021 年 3 月 18 日

AWS ParallelCluster 2.10.2 版已發行	AWS ParallelCluster 2.10.2 版已發行。 變更包括： • 改善叢集組態驗證，以在 `--dry-run` 模式下叫用 Amazon EC2RunInstancesAPI操作AMI時使用叢集目標。 • 將 AWS ParallelCluster 虛擬環境中使用的 Python 版本更新為 3.6.13。 • `sanity_check` Arm 執行個體類型的修正。 • `centos8` 搭配 使用 `enable_efa` 時修正 Slurm 排程器或 Arm 執行個體類型。 • `apt update` 以非互動式模式執行（`-y`）。 • Fix `encrypted_ephemeral` = 使用 `alinux2`和 為 `truecentos8`。 如需變更的詳細資訊，請參閱上的 aws-parallelcluster 套件 CHANGELOG檔案 GitHub。	2021 年 3 月 2 日

AWS ParallelCluster 2.10.1 版已發行	AWS ParallelCluster 2.10.1 版已發行。
	
	變更包括：
	
	• 新增對非洲（開普敦）（）af-south-1 、歐洲（米蘭）（me-south-1 ）和中東（巴林）（me-south-1 ）的支援 AWS 區域。啟動時，支援會受到下列方式的限制：
	  • FSx 任何這些 都不支援 for Lustre 和 Arm 型 Graviton 執行個體 AWS 區域。
	  • AWS Batch 非洲（開普敦）不支援 。
	  • 非洲（開普敦）EBSio2和歐洲（米蘭）不支援 Amazon 和gp3磁碟區類型 AWS 區域。
	• 已新增對 Amazon EBSio2和gp3磁碟區類型的支援。如需詳細資訊，請參閱[ebs]章節 和[raid]章節 。
	• 新增對執行 alinux2、ubuntu1804 或 Elastic Fabric Adapter之 Arm 型 Graviton2 執行個體的支援ubuntu2004 。如需詳細資訊，請參閱Elastic Fabric Adapter。
		2020 年 12 月 22 日

- 在 Arm AMIs（`alinux2`、`centos8`和 ）上安裝 Arm Performance 程式庫 `20.2.1ubuntu1804` 。如需詳細資訊，請參閱Arm Performance 程式庫。
- Intel MPI 已更新至 2019 版更新 8（更新自 2019 版更新 7）。如需詳細資訊，請參閱 Intel® MPI Library 2019 Update 8。
- 從 AWS Batch Docker 進入點移除 AWS CloudFormation `DescribeStacks` API操作呼叫，以結束 調節所導致的任務失敗 AWS CloudFormation。
- 驗證叢集組態時，改善對 Amazon EC2`DescribeInstanceTypes` API操作呼叫的呼叫。
- Amazon Linux 2 Docker 映像會在建置`awsbatch`排程器的 Docker 映像時，從 Amazon ECR Public 擷取。
- 的預設執行個體類型從硬式編碼`t2.micro`執行個體類型變更為 AWS 區域（`t2.micro` 或的 Free Tier 執行個體類型`t3.micro`，取決於 AWS 區域）AWS 區域 。 沒有執行個體`t3.micro`類型的 Free Tier 預設值。

- Elastic Fabric Adapter 安裝程式已更新至 1.11.1：
  - EFA 組態：efa-config-1.7（來自 efa-config-1.5）
  - EFA 設定檔：efa-profile-1.3（來自 efa-profile-1.1）
  - EFA 核心模組：efa-1.10.2（無變更）
  - RDMA 核心：rdma-core-31.2amzn（來自 rdma-core-31.amzn0）
  - Libfabric：libfabric-1.11.1amzn1.0（來自 libfabric-1.10.1amzn1.1）
  - 開啟 MPI：openmpi40-aws-4.1.0（從 openmpi40-aws-4.0.5）
- 現在需要 vpc_settings、vpc_id和 master_subnet_id 參數。
- 主節點中的常駐nfsd程式現在設定為使用至少 8 個執行緒。如果有超過 8 個核心，則會使用與核心一樣多的執行緒。使用 ubuntu1604 時，設定只會在節點重新啟動後變更。

- Amazon DCV 已更新為 Amazon DCV 2020.2-9662。如需詳細資訊，請參閱 Amazon DCV管理員指南中的 DCV 2020.2-9662 — 2020 年 12 月 4 日。
- 的 Intel MPI和 HPC套件是從 Amazon S3 AWS ParallelCluster 提取。它們不再從 Intel yum repos 中提取。
- 已變更預設值 systemd 正式 建立OSs期間執行層級 `multi-user.target` 至 AWS ParallelCluster AMIs。只有在啟用時，執行層級才會在主節點`graphical.target` 上設定為 DCV 。這可防止圖形服務（例如 `x/gdm`）在不需要時執行。
- 已啟用對主機節點上 `p4d.24xlarge` 執行個體的支援。
- 增加註冊時的重試嘗試次數上限 Slurm Amazon Route 53 中的節點。

如需變更的詳細資訊，請參閱 上的 aws-parallelcluster、 aws-parallelcluster-cookbook和 aws-parallelcluster-node套件CHANGELOG的檔案 GitHub。

AWS ParallelCluster 2.10.0 版已發行	AWS ParallelCluster 2.10.0 版已發行。
	
	變更包括：
	
	• 新增 的支援 CentOS 所有 AWS 區域（ AWS 中國和 AWS GovCloud（美國）區域以外）為 8。已移除對 的支援 CentOS 6.
	• 已新增對運算節點`p4d.24xlarge` 執行個體的支援。
	• EFA 使用新`enable_efa_gdr` 設定新增對 NVIDIA GPUDirect RDMA 的支援。
	• 新增對 Amazon FSx for Lustre 功能的支援。
	  • 設定您的 Amazon FSx for Lustre 檔案系統，以使用`auto_import_policy` 設定匯入偏好設定。
	  • 已使用 `storage_type` 和 `drive_cache_type` 設定新增對 HDD型 Amazon FSx for Lustre 檔案系統的支援。
	• 新增 Amazon CloudWatch 儀表板，包括頭節點指標和輕鬆存取叢集日誌。如需詳細資訊，請參閱Amazon CloudWatch 儀表板。
	• 新增使用 `cluster_resource_bucket` 設定使用現有 Amazon S3 儲存貯
		2020 年 11 月 18 日

體來存放叢集組態資訊的支援。

- 增強 pcluster createami 命令。
  - 新增--post-install 參數，以在建置 時使用安裝後指令碼 AMI。
  - 新增驗證步驟，以便在使用不同 版本AMI建立的基礎時失敗 AWS ParallelCluster。
  - 如果選取的作業系統與基礎 中的作業系統不同，則新增驗證步驟失敗AMI。
  - 新增了對使用 AWS ParallelCluster 基礎 的支援AMI。
- 增強 pcluster update命令。
  - 現在可以在更新期間變更tags設定。
  - 現在可以在更新期間調整佇列的大小，而無需停止運算機群
- 新增slurm_resume 指令碼的all_or_nothing_batch 組態參數。當 時True，只有在 中所有待定任務所需的所有執行個體都會slurm_resume 成功 Slurm 將可供使用。如需詳細資訊，請參閱在 上的 AWS ParallelCluster Wiki 中介紹all_or_no

`thing_batch` 啟動 GitHub。

- Elastic Fabric Adapter 安裝程式已更新至 1.10.1：
  - EFA 組態：`efa-config-1.5`（來自 `efa-config-1.4`）
  - EFA 設定檔：`efa-profile-1.1`（來自 `efa-profile-1.0.0`）
  - EFA 核心模組：`efa-1.10.2`（來自 `efa-1.6.0`）
  - RDMA 核心：`rdma-core-31.amzn0`（來自 `rdma-core-28.amzn0`）
  - Libfabric：`libfabric-1.11.1amzn1.0`（來自 `libfabric-1.10.1amzn1.1`）
  - 開啟 MPI：`openmpi40-aws-4.0.5`（從 `openmpi40-aws-4.0.3`）
- 在 AWS GovCloud (US) 區域中，啟用 Amazon DCV和的支援 AWS Batch。
- AWS 在中國區域中，啟用對 Amazon FSx for Lustre 的支援。

- 將NVIDIA驅動程式升級至 450.80.02 版（從 450.51.05）。
- 安裝 NVIDIA Fabric Manager 在NVIDIANV Switch支援的平台上啟用 。
- 已移除 AWS 區域 的預設值us-east-1 。預設值會使用此查詢順序。
  - AWS 區域 在 -r或 --region引數中指定。
  - AWS_DEFAULT_REGION 環境變數。
  - aws_region_name [aws] AWS ParallelCluster 設定（預設為 ~/.parallelcluster/config ）。
  - region [default] AWS CLI 設定（預設為 ~/aws/config ）。

如需變更的詳細資訊，請參閱 上的 aws-parallelcluster、 aws-parallelcluster-cookbook和 aws-parallelcluster-node套件CHANGELOG的檔案 GitHub。

AWS ParallelCluster 2.9.0 版已發行	AWS ParallelCluster 2.9.0 版已發行。 變更包括： • 與 搭配使用時，已新增對運算機群中多個佇列和多個執行個體類型的支援 Slurm Workload Manager。 使用佇列時，Auto Scaling 群組不再用於 Slurm。 Amazon Route 53 託管區域現在已使用叢集建立，並在 DNS Slurm 排程器已使用。如需詳細資訊，請參閱多重佇列模式。 • 新增對 Amazon DCV on Arm 型 AWS Graviton 執行個體的支援。 • 新增了對不支援啟動範本（例如執行個體類型）中CPU選項之*.metal執行個體類型的停用超執行緒的支援。 • 已新增對從主機節點共用之檔案系統的 NFS4 支援。 • 在引導運算節點時，移除對 cfn-init 的依賴，以避免大量節點加入叢集 AWS CloudFormation 時調節。 • Elastic Fabric Adapter 安裝程式已更新至 1.9.5： • EFA 組態：efa-config-1.4 （來自 efa-config-1.3 ）	2020 年 9 月 11 日

    - EFA 設定檔：efa-profile-1.0.0 （新）
    - 核心模組：efa-1.6.0 (未變更)
    - RDMA 核心：rdma-core-28.amzn0 （無變更）
    - Libfabric：libfabric-1.10.1amzn1.1 (未變更)
    - 開啟 MPI： openmpi40-aws-4.0.3 （無變更）
- 已升級 Slurm 至 版本 20.02.4（從 19.05.5）。
- Amazon DCV 已更新至 Amazon DCV 2020.1-9012。如需詳細資訊，請參閱 Amazon DCV管理員指南 中的 DCV 2020.1-9012 — 2020 年 8 月 24 日版本備註。
- 安裝共用NFS磁碟機時，請使用主機節點私有 IP 地址，而非主機名稱。
- 已將新的日誌串流新增至 CloudWatch 日誌：chef-client 、clustermgtd 、slurm_resume 、 computemgtd 和 slurm_suspend 。
- 新增了對預先安裝和安裝後指令碼中佇列名稱的支援。
- 在 中 AWS GovCloud (US) AWS 區域，使用 Amazon

	DynamoDB 隨需計費選項。如需詳細資訊，請參閱 Amazon DynamoDB 開發人員指南 中的隨需模式。 如需變更的詳細資訊，請參閱 上的 aws-parallelcluster、 aws-parallelcluster-cookbook和 aws-parallelcluster-node套件CHANGELOG的檔案 GitHub。	
AWS ParallelCluster 2.8.1 版已發行	AWS ParallelCluster 2.8.1 版已發行。 變更包括： • 停用 Amazon DCV工作階段的螢幕鎖定，以防止使用者遭到鎖定。 • 在包含 Arm 型 AWS Graviton 型執行個體類型`pcluster configure`時修正 。 如需變更的詳細資訊，請參閱 上的 aws-parallelcluster、 aws-parallelcluster-cookbook和 aws-parallelcluster-node套件CHANGELOG的檔案 GitHub。	2020 年 8 月 4 日

AWS ParallelCluster 2.8.0 版已發行	AWS ParallelCluster 2.8.0 版已發行。
	
	變更包括：
	
	• 新增對 Arm 型 AWS Graviton 型執行個體的支援（例如 A1和 C6g）。
	• 新增對 Amazon FSx for Lustre 自動每日備份功能的支援。如需詳細資訊，請參閱 automatic_backup_retention_days 、copy_tags_to_backups 、daily_automatic_backup_start_time 和 fsx_backup_id 。
	• 從 移除對 Berkshelf 的相依性pcluster createami_。
	• 改善了 的穩健性和使用者體驗pcluster update。如需詳細資訊，請參閱使用 pcluster update。
	• Elastic Fabric Adapter 安裝程式已更新至 1.9.4：
	  • 核心模組： efa-1.6.0 （更新自 efa-1.5.1 ）
	  • RDMA 核心： rdma-core-28.amzn0 （更新自 rdma-core-25.0 ）
		2020 年 7 月 23 日

    - Libfabric：`libfabric-1.10.1amzn1.1`（更新自 `libfabric-aws-1.9.0amzn1.1`）
    - 開啟 MPI：`openmpi40-aws-4.0.3`（無變更）
- 在 上將NVIDIA驅動程式升級至 Tesla 版本 440.95.01 CentOS 第 6 版和第 450.51.05 版。
- 在除 以外的所有分發上將CUDA程式庫升級至 11.0 版 CentOS 6.

如需變更的詳細資訊，請參閱 上的 aws-parallelcluster、aws-parallelcluster-cookbook和 aws-parallelcluster-node套件CHANGELOG的檔案 GitHub。

AWS ParallelCluster 2.7.0 版已發行	AWS ParallelCluster 2.7.0 版已發行。 變更包括： • base_os 現在是必要參數。 • scheduler 現在是必要參數。 • Amazon DCV 已更新至 Amazon DCV 2020.0。如需詳細資訊，請參閱 Amazon DCV2020.0 版搭配環繞音效 7.1 和觸控筆支援。 Intel MPI 已更新至 2019 版更新 7（更新自 2019 版更新 6）。如需詳細資訊，請參閱 Intel® MPI Library 2019 Update 7。 更新 Elastic Fabric Adapter 安裝程式到 1.8.4： • 核心模組：efa-1.5.1 (未變更) • RDMA 核心：rdma-core-25.0（無變更） • Libfabric：libfabric-aws-1.9.0amzn1.1 (未變更) • 開啟 MPI：openmpi40-aws-4.0.3（更新自 openmpi40-aws-4.0.2） • 升級 CentOS 7 AMI至 7.8-2003 版（更新自 7.7-1908）。如需詳細	2020 年 5 月 19 日

	資訊，請參閱 CentOS-7（2003）版本備註。	
AWS ParallelCluster 2.6.1 版已發行	AWS ParallelCluster 2.6.1 版已發行。 變更包括： • `cfn-wire` 已從存放在 Amazon CloudWatch Logs 中的日誌中移除 `cfn-init-cmd` 和 。如需詳細資訊，請參閱與 Amazon CloudWatch Logs 整合。	2020 年 4 月 17 日

AWS ParallelCluster 2.6.0 版已發行	AWS ParallelCluster 2.6.0 版已發行。 變更包括： • 已新增 Amazon Linux 2 的支援。 • 現在，Amazon CloudWatch Logs 用於收集叢集和排程器日誌。如需詳細資訊，請參閱與 Amazon CloudWatch Logs 整合。 • 新增對新的 Amazon FSx for Lustre 部署類型 `SCRATCH_2` 和 的支援`PERSISTENT_1` 。FSx 支援 上的 Lustre Ubuntu 18.04 和 Ubuntu 16.04. 如需詳細資訊，請參閱：fsx。 • 新增對 Amazon DCV on 的支援 Ubuntu 18.04. 如需詳細資訊，請參閱透過 Amazon 連線至主機節點 DCV。	2020 年 2 月 27 日
AWS ParallelCluster 2.5.1 版已發行	AWS ParallelCluster 2.5.1 版已發行。	2019 年 12 月 13 日
AWS ParallelCluster 2.5.0 版已發行	AWS ParallelCluster 2.5.0 版已發行。	2019 年 11 月 18 日
AWS ParallelCluster 推出對 Intel 的支援 MPI	AWS ParallelCluster 2.4.1 版推出對 Intel 的支援。 MPI	2019 年 7 月 29 日
AWS ParallelCluster 引入 的支援 EFA	AWS ParallelCluster 2.4.0 版推出對 Elastic Fabric Adapter（EFA）的支援。	2019 年 6 月 11 日

AWS ParallelCluster 文件網站上發佈 AWS 的文件	AWS ParallelCluster 文件現在提供 10 種語言，以及 HTML和 PDF 格式。	2018 年 5 月 24 日

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。