



AWS ParallelCluster 使用者指南 (v3)

# AWS ParallelCluster



# AWS ParallelCluster: AWS ParallelCluster 使用者指南 (v3)

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

AWS ParallelCluster 是什麼 .....	1
定價 .....	1
設定 AWS ParallelCluster .....	2
設置一個 AWS 帳戶 .....	2
註冊一個 AWS 帳戶 .....	2
建立具有管理權限的使用者 .....	3
建立金鑰對 .....	4
安裝 AWS ParallelCluster CLI 的安裝 .....	4
AWS ParallelCluster 在虛擬環境中安裝 (建議使用) .....	4
使用 pip AWS ParallelCluster 在非虛擬環境中安裝 .....	6
安裝 AWS ParallelCluster 為獨立應用程式 .....	7
安裝後要採取的步驟 .....	9
安裝AWS ParallelCluster使用者介面 .....	9
安裝AWS ParallelCluster使用者介面 .....	10
建立自訂網域 .....	12
Amazon Cognito 使用者集區選項 .....	14
識別AWS ParallelCluster和 AWS ParallelCluster UI 版本 .....	17
將 AWS ParallelCluster UI 更新為新AWS ParallelCluster版本 .....	17
AWS ParallelClusterUI 成本 .....	17
開始使用 .....	18
使用 AWS ParallelCluster CLI 設定和建立叢集 .....	18
使 AWS ParallelCluster 用 UI 配置和創建集群 .....	27
連接至叢集 .....	28
多個使用者存取叢集 .....	29
創建一個活動目錄 .....	30
建立具有 AD 網域的叢集 .....	30
登入與 AD 網域整合的叢集 .....	33
執行 MPI 工作 .....	34
AWS Managed Microsoft AD 透過 LDAP (S) 叢集配置的範例 .....	34
最佳實務 .....	38
最佳實踐：頭節點實例類型選擇 .....	38
最佳做法：網路效能 .....	38
最佳做法：預算提醒 .....	40
最佳做法：將叢集移至新的次要版本或 AWS ParallelCluster 修補程式版本 .....	40

從 AWS ParallelCluster 2.x 移動到 3.x .....	41
自訂引導操作 .....	41
AWS ParallelCluster 2.x 和 3.x 使用不同的配置文件語法 .....	41
包容性語言 .....	47
排程器 Support .....	48
AWS ParallelCluster CLI .....	48
IMDS 組態設定更新 .....	51
AWS ParallelCluster 支援的區域 .....	51
使用 AWS ParallelCluster .....	53
AWS ParallelCluster 界面 .....	54
AWS Lambda 中的 VPC 組態 AWS ParallelCluster .....	55
AWS Identity and Access Management 權限 AWS ParallelCluster .....	56
AWS ParallelCluster Amazon EC2 實例角色 .....	57
AWS ParallelCluster <code>pccluster</code> 使用者策略範例 .....	58
AWS ParallelCluster 管理 IAM 資源的使用者範例政策 .....	72
AWS ParallelCluster 用於管理 IAM 許可的組態參數 .....	78
網路組態 .....	92
AWS ParallelCluster 在單一公用子網路中 .....	93
AWS ParallelCluster 使用兩個子網 .....	95
AWS ParallelCluster 在使用連接的單個私有子網中 AWS Direct Connect .....	96
AWS ParallelCluster 與 AWS Batch 調度程序 .....	97
AWS ParallelCluster 在沒有互聯網訪問的單個子網中 .....	99
登入節點 .....	104
自訂引導動作 .....	107
組態 .....	109
引數 .....	112
包含自訂引導動作的範例叢集 .....	112
更新 IMDSv2 的自訂啟動程序指令碼範例 .....	114
更新 IMDSv1 組態的範例 .....	115
使用 Amazon S3 .....	116
範例 .....	116
使用 競價型執行個體 .....	117
案例 1：沒有執行中任務的 Spot 執行個體遭到中斷 .....	117
案例 2：執行單一節點任務的 Spot 執行個體遭到中斷 .....	117
案例 3：執行多節點任務的 Spot 執行個體遭到中斷 .....	118
排程器支援 AWS ParallelCluster .....	118

Slurm Workload Manager .....	118
AWS Batch .....	174
共用儲存 .....	182
設定共用儲存 .....	184
使用共用儲存裝置 .....	186
配額 .....	190
標記 .....	190
監控 AWS ParallelCluster 和記錄 .....	193
與 Amazon Amazon Amazon CloudWatch 的整合 .....	194
Amazon CloudWatch 儀表 .....	198
針對叢集指標的 Amazon CloudWatch 警示 .....	200
AWS ParallelCluster配置的日誌輪替 .....	203
pclusterCLI 記錄檔 .....	204
Amazon EC2 主控台輸出日誌 .....	204
擷取AWS ParallelCluster UI 和AWS ParallelCluster執行階段記錄 .....	205
擷取和保留記錄 .....	207
AWS CloudFormation 自訂資源 .....	210
提供者堆疊由 AWS ParallelCluster .....	210
叢集資源 .....	212
叢集作業 .....	215
疑難排解包含 AWS ParallelCluster 自訂資源的堆疊 .....	215
Elastic Fabric Adapter .....	216
啟用英特爾 MPI .....	217
AWS ParallelCluster API .....	218
AWS ParallelCluster API 文件 .....	218
使用部署 AWS CLI .....	219
正在更新 API .....	221
調用 API AWS ParallelCluster .....	222
訪問 API 日誌和指標 .....	224
AWS ParallelCluster 對於地形 .....	224
通過 NICE DCV Connect 到頭節點 .....	225
NICE DCV DCV HTTPS 證書 .....	226
NICE DCV 發牌 .....	226
使用 pcluster update-cluster .....	226
更新原則：定義 .....	226
pcluster update-cluster 範例 .....	229

AWS ParallelCluster AMI 定制 .....	232
AWS ParallelCluster AMI 自訂考量 .....	232
執行自訂元件驗證測試 .....	233
使用pcluster指令監視 Image Builder 程序，以協助偵錯 .....	233
其他考量 .....	234
使用隨需容量保留 (ODCR) 啟動執行個體 .....	234
使用 OCR 搭配使用 AWS ParallelCluster .....	235
使用容量區塊 (CB) 啟動執行個體 .....	243
使用 CB 搭配使用 AWS ParallelCluster .....	243
AMI 修補和 Amazon EC2 實例替換 .....	244
頭節點實例更新或更換 .....	245
從暫時驅動器保存數據 .....	245
停止並啟動叢集的頭節點 .....	246
作業系統 .....	247
作業系統考量 .....	248
參考資料 AWS ParallelCluster .....	250
AWS ParallelCluster第 3 版 CLI 指令 .....	250
pcluster .....	251
pcluster3-config-converter .....	293
組態檔案 .....	294
叢集配置檔 .....	294
構建映像配置文件 .....	411
AWS ParallelCluster API 參考資料 .....	419
建造影像 .....	420
createCluster .....	425
刪除叢集 .....	431
deleteClusterInstances .....	434
刪除影像 .....	436
描述叢集 .....	439
describeClusterInstances .....	447
describeComputeFleet .....	450
描述圖片 .....	452
getClusterLog活動 .....	459
getClusterStack活動 .....	463
getImageLog活動 .....	467
getImageStack活動 .....	471

列表集群 .....	475
listClusterLog串流 .....	479
listImageLog串流 .....	483
列表法師 .....	487
listOfficialImages .....	490
更新叢集 .....	493
updateComputeFleet .....	499
AWS ParallelClusterPython 函式庫 .....	501
AWS ParallelCluster程式庫授權 .....	502
安裝AWS ParallelCluster蟒蛇圖書館 .....	502
叢集 API 作業 .....	502
運算叢集 API 作業 .....	506
叢集和堆疊記錄作業 .....	508
影像 API 作業 .....	511
影像和堆疊記錄作業 .....	513
範例 .....	516
AWS Lambda為了AWS ParallelCluster蟒蛇圖書館 .....	517
AWS ParallelCluster 的運作方式 .....	519
AWS ParallelCluster 程序 .....	519
clustermgtd .....	519
clusterstatusmgtd .....	520
computemgtd .....	520
AWS 使用的服務 AWS ParallelCluster .....	520
Amazon API Gateway .....	521
AWS Batch .....	522
AWS CloudFormation .....	522
Amazon CloudWatch .....	522
Amazon CloudWatch 活動 .....	522
Amazon CloudWatch 日誌 .....	523
AWS CodeBuild .....	523
Amazon DynamoDB .....	523
Amazon Elastic Block Store .....	523
Amazon Elastic Compute Cloud .....	524
Amazon Elastic Container Registry .....	524
Amazon EFS .....	524
Amazon FSx for Lustre .....	524

Amazon FSx NetApp .....	525
Amazon FSx for OpenZFS .....	525
AWS Identity and Access Management .....	525
AWS Lambda .....	525
Amazon RDS .....	526
Amazon Route 53 .....	526
Amazon Simple Notification Service .....	526
Amazon Simple Storage Service .....	526
Amazon VPC .....	527
Elastic Fabric Adapter .....	527
EC2 Image Builder .....	527
NICE DCV .....	527
AWS ParallelCluster內部目錄 .....	527
教學課程 .....	529
執行您的第一份工作 AWS ParallelCluster .....	529
確認安裝 .....	530
建立您的第一個叢集 .....	530
登錄到您的頭節點 .....	531
使用思慮姆運行你的第一份工作 .....	532
構建自定義 AWS ParallelCluster AMI .....	533
如何自定義 AWS ParallelCluster AMI .....	534
建立自訂 AWS ParallelCluster AMI .....	534
修改 AWS ParallelCluster AMI .....	540
整合作用中目錄 .....	542
使用 AWS KMS 金鑰設定共用儲存區加密 .....	573
建立 政策 .....	574
配置和建立叢集 .....	574
在多佇列模式叢集中執行工作 .....	576
配置您的叢集 .....	576
建立 叢集 .....	578
登入頭節點 .....	579
以多重佇列模式執行工作 .....	579
應用 AWS ParallelCluster 程式介面 .....	583
使用Slurm會計建立叢集 .....	597
步驟 1：建立下列項目的 VPC 和子網路 AWS ParallelCluster .....	598
步驟 2：建立資料庫堆疊 .....	598



步驟 3：建立啟用Slurm會計功能的叢集 .....	598
使用外部Slurmdbd帳戶建立叢集 .....	599
第 1 步：創建思路堆棧 .....	600
步驟 2：建立Slurmdbd啟用外部的叢集 .....	601
還原至先前的AWS系統管理員文件版本 .....	602
還原為先前的 SSM 文件版本 .....	602
建立叢集 AWS CloudFormation .....	604
使用 CloudFormation 快速建立堆疊建立叢集 .....	604
使用 AWS CloudFormation 命令列介面 (CLI) 建立叢集 .....	607
檢視 CloudFormation 叢集輸出 .....	609
存取您的叢集 .....	609
清除 .....	610
使用地形部署 ParallelCluster API .....	610
定義地形專案 .....	610
部署 API .....	612
所需的許可 .....	613
使用地形建立叢集 .....	616
定義地形專案 .....	616
部署叢集 .....	622
所需的許可 .....	624
使用地形創建自定義 AMI .....	624
定義地形專案 .....	625
部署 AMI .....	628
所需的許可 .....	628
AWS ParallelCluster與身分識別中心的 UI 整合 .....	629
啟用 IAM Identity Center .....	629
將您的應用程式新增至 IAM 身分中心 .....	632
AWS ParallelCluster 疑難排 .....	640
嘗試建立叢集 .....	641
failureCode 是 OnNodeConfiguredExecutionFailure .....	641
failureCode 是 OnNodeConfiguredDownloadFailure .....	641
failureCode 是 OnNodeConfiguredFailure .....	642
failureCode 是 OnNodeStartExecutionFailure .....	642
failureCode 是 OnNodeStartDownloadFailure .....	642
failureCode 是 OnNodeStartFailure .....	643
failureCode 是 EbsMountFailure .....	643

failureCode 是 EfsMountFailure .....	643
failureCode 是 FsxMountFailure .....	644
failureCode 是 RaidMountFailure .....	644
failureCode 是 AmiVersionMismatch .....	644
failureCode 是 InvalidAmi .....	644
failureCodefailureReason無法設置頭節點。HeadNodeBootstrapFailure .....	645
failureCodefailureReason叢集建立逾時。HeadNodeBootstrapFailure .....	645
failureCodefailureReason無法啟動頭節點。HeadNodeBootstrapFailure .....	646
failureCode 是 ResourceCreationFailure .....	647
failureCode 是 ClusterCreationFailure .....	647
看到 CloudFormation , WaitCondition timed out... 在, 堆 .....	647
看到 CloudFormation , Resource creation cancelled 在, 堆 .....	647
查看Failed to run cfn-init...或 AWS CloudFormation 堆棧中的其他錯誤 .....	647
看到chef-client.log結束 INFO: Waiting for static fleet capacity provisioning .....	648
看得見 Failed to run preinstall or postinstall in cfn-init.log .....	648
看到 CloudFormation , This AMI was created with xxx, but is trying to be used with xxx... 在, 堆 .....	648
看到 CloudFormation , This AMI was not baked by AWS ParallelCluster... 在, 堆 .....	648
看到pcluster create-cluster命令無法在本地運行 .....	648
其他支援 .....	648
嘗試執行工作 .....	649
srun互動式工作因錯誤而失敗 srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf .....	649
使用queue命令將 Job 停留在CF狀態 .....	649
執行大規模工作並查看 nfsd: too many open connections, consider increasing the number of threads in /var/log/messages .....	649
執行 MPI 工作 .....	650
嘗試更新叢集 .....	651
pcluster update-cluster命令無法在本地運行 .....	651
看到clusterStatus是UPDATE_FAILED用pcluster describe-cluster命令 .....	651
叢集更新逾時 .....	651
嘗試訪問存儲 .....	651
使用外部 Amazon FSx for Lustre 文件系統 .....	651
使用外部 Amazon Elastic File System 檔案系統 .....	651

嘗試刪除叢集 .....	651
該pcluster delete-cluster命令無法在本地運行 .....	651
叢集堆疊無法刪除 .....	652
嘗試升級 AWS ParallelCluster API 堆棧 .....	652
在計算節點初始化中看到錯誤 .....	652
看到Node bootstrap error在 clustermgtd.log .....	652
我設定了隨需容量保留 (ODC) 或區域預留執行個體 .....	652
An error occurred (VcpuLimitExceeded)在我無法運行作業slurm_resume.log時 看到，或者在clustermgtd.log我無法創建集群時看到 .....	653
An error occurred (InsufficientInstanceCapacity)在我無法運行作 業slurm_resume.log時看到，或者在clustermgtd.log我無法創建集群時看到 .....	654
查看節點DOWN處於狀態 Reason (Code:InsufficientInstanceCapacity)... .....	654
看到cannot change locale (en_US.utf-8) because it has an invalid name在 slurm_resume.log .....	654
以前的情況都不適用於我的情況 .....	655
疑難排解叢集健康度 .....	655
查看執行處理啟動設定錯誤圖 .....	655
查看狀態不良的執行個體錯誤圖表 .....	657
查看運算叢集閒置時間圖表 .....	658
疑難排解叢集部署問 .....	659
查看 AWS CloudFormation 事件 CREATE_FAILED .....	660
使用 CLI 檢視記錄串流 .....	662
使用重新建立失敗的叢集 rollback-on-failure .....	664
使用地形進行叢集部署疑難排解 .....	665
ParallelCluster 找不到 API .....	665
未授權呼叫 ParallelCluster API 的使用者 .....	665
排解擴展問題 .....	666
調試的關鍵日誌 .....	667
slurm_resume.log當我無法運行作業或無法創建集群clustermgtd.log時看 到InsufficientInstanceCapacity錯誤 .....	654
解決節點初始化問題 .....	669
疑難排解非預期的節點取代和終止 .....	671
取代、終止或關閉有問題的執行個體和節點 .....	673
佇列 ( 分割區 ) Inactive狀態 .....	673
疑難排解其他已知節點和工作問題 .....	673
放置群組和執行個體啟動問題 .....	673

無法取代的目錄 .....	674
疑難排解 NICE DCV 中的問題 .....	674
NICE DCV DCV 的日誌 .....	674
NICE DCV DCV 問題 .....	674
使用整合進行叢 AWS Batch 集的疑難排解 .....	675
頭節點問題 .....	675
運算問題 .....	675
Job 失敗 .....	675
端點 URL 上的 Connect 逾時錯誤 .....	676
疑難排解多使用者與活動目錄整合 .....	676
活動目錄特定的故障 .....	677
啟用除錯模式 .....	677
如何從網路轉移到 LDAP .....	678
如何停用 LDAPS 伺服器憑證驗證 .....	678
如何使用 SSH 密鑰而不是密碼登錄 .....	678
如何重置用戶密碼和過期的密碼 .....	678
如何驗證加入的網域 .....	679
如何疑難排解憑證問題 .....	680
如何驗證與活動目錄的集成是否正常工作 .....	681
如何疑難排解登入運算節點的問題 .....	682
在多使用者環境中， SimCenter StarCCM+ 工作的已知問題 .....	682
用戶名解析的已知問題 .....	683
如何解決主目錄創建問題 .....	683
排解自訂 AMI 問題 .....	684
疑難排解 cfn-hup 未執行時的叢集更新逾時 .....	685
網路疑難排 .....	685
單一公用子網路中的叢集問題 .....	685
onNodeUpdated 自訂動作的叢集更新失敗 .....	686
查看自定義配 Slurm 置的錯誤 .....	686
叢集警示 .....	686
其他支援 .....	687
AWS ParallelCluster 支援政策 .....	688
安全 .....	689
所使用服務的安全性資訊 AWS ParallelCluster .....	689
資料保護 .....	690
資料加密 .....	691

---

另請參閱 .....	691
Identity and Access Management .....	692
合規驗證 .....	692
強制執行 TLS 1.2 .....	693
判定目前支援的通訊協定 .....	693
編譯 OpenSSL 和 Python .....	695
版本說明和文件記錄 .....	697
.....	dcclxxxi

# AWS ParallelCluster 是什麼

AWS ParallelCluster是AWS支援的開放原始碼叢集管理工具，可協助您部署和管理 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。

您可以使AWS ParallelCluster用下列方法存取：

- [AWS ParallelCluster命令行界面](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster用戶界面](#) ( 添加了 3.5.0 版本 )
- [AWS ParallelClusterPython 函式庫](#) ( 與版本 3.5.0 一起添加 )
- 作為[AWS CloudFormation 自訂資源](#) ( 添加了版本 3.6.0 )

## 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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelClusterUI 建立在無伺服器架構上，在大多數情況下，您可以在AWS免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

# 設定 AWS ParallelCluster

## 主題

- [設置一個 AWS 帳戶](#)
- [建立金鑰對](#)
- [安裝 AWS ParallelCluster 命令列介面 \(CLI\)](#)
- [安裝後要採取的步驟](#)
- [安裝 AWS ParallelCluster 使用者介面](#)
- [開始使用 AWS ParallelCluster](#)
- [多個使用者存取叢集](#)
- [最佳實務](#)
- [從 AWS ParallelCluster 2.x 移動到 3.x](#)
- [AWS ParallelCluster 支援的區域](#)

## 設置一個 AWS 帳戶

設定要使用的 AWS 帳戶 AWS ParallelCluster。

### 註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

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

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建 AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 <https://aws.amazon.com/> 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

## 建立具有管理權限的使用者

註冊後，請保護您的 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

### 保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。[AWS Management Console](#)在下一頁中，輸入您的密碼。

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

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

### 建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM 身分中心中，將管理存取權授予使用者。

[若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。](#)

### 以具有管理權限的使用者身分登入

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

如需使用 IAM 身分中心使用者[登入的說明，請參閱使用AWS 登入 者指南中的登入 AWS 存取入口網站](#)。

### 指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[建立權限集](#)」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。



如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[新增群組](#)」。

## 建立金鑰對

若要部署叢集，請 AWS ParallelCluster 啟動 Amazon EC2 執行個體以建立叢集主節點和運算節點。若要執行叢集工作 (例如執行和監視作業) 或管理使用者，您必須能夠存取叢集標頭節點。若要驗證您可以使用 SSH 存取頭節點執行個體，您必須使用 Amazon EC2 key pair。若要了解如何建立 key pair，請參閱 Amazon 彈性運算雲端 Linux 執行個體使用者指南中的[建立 key pair](#)。

## 安裝命 AWS ParallelCluster 令列介面 (CLI)

AWS ParallelCluster 作為 Python 軟件包進行分發，並使用 Python 軟 pip 件包管理器進行安裝。如需有關如何安裝 Python 套件的指示，請參閱 Python [封裝](#) 使用者指南中的安裝套件。

安裝方式 AWS ParallelCluster :

- [AWS ParallelCluster 在虛擬環境中安裝 \(建議使用\)](#)
- [使用 pip AWS ParallelCluster 在非虛擬環境中安裝](#)
- [安裝 AWS ParallelCluster 為獨立應用程式](#)

您可以在上的 [\[發行版本\] 頁面](#) 上找到最新 CLI 的版本號碼 GitHub。在本指南中，指令範例假設您已安裝的 Python 版本低於 3.6 版。pip 命令範例使用的是 pip3 版本。

管理 AWS ParallelCluster 2 和 AWS ParallelCluster 3

對於同時使用 AWS ParallelCluster 2 和 AWS ParallelCluster 3 且想要管理這兩個套件之 CLI 的客戶，我們建議您在不同的 [虛擬環境](#) 中安裝 AWS ParallelCluster 2 和 AWS ParallelCluster 3。這可確保您可以繼續使用的每個版本以 AWS ParallelCluster 及任何關聯的叢集資源。

## AWS ParallelCluster 在虛擬環境中安裝 (建議使用)

我們建議您在虛擬環境 AWS ParallelCluster 中進行安裝，以避免與其他 pip 套件的需求版本衝突。

必要條件

- AWS ParallelCluster 需要 Python 或更高版本。如果您尚未安裝它，請在 [pyth on.org](#) [下載適用於您平台的兼容版本](#)。

## 在虛擬環境 AWS ParallelCluster 中安裝

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

```
$ 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。

2. 建立虛擬環境並為其命名。

```
$ 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
```

3. 啟用新的虛擬環境。

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

4. 安裝 AWS ParallelCluster 到您的虛擬環境中。

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

5. 安裝節點版本管理員和最新的長期 Support (LTS) Node.js 版本。AWS Cloud Development Kit (AWS CDK) (AWS CDK) 需要 Node.js 才能產 CloudFormation 生範本。

### Note

如果您的 Node.js 安裝無法在您的平台上運行，則可以在最新的 LTS 版本之前安裝 LTS 版本。如需詳細資訊，請參閱 [Node.js 版本排程](#) 和 [AWS CDK](#) 先決條件。

Node.js 安裝命令範例：

```
$ nvm install --lts=Hydrogen
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

```
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

## 6. 確認已 AWS ParallelCluster 正確安裝。

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

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](#)。

若要升級到最新版本的 AWS ParallelCluster，請再次執行安裝命令。

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

## 使用 pip AWS ParallelCluster 在非虛擬環境中安裝

### 必要條件

- AWS ParallelCluster 需要 Python 或更高版本。如果您尚未安裝它，請在 [python.org](https://python.org) [下載適用於您平台的兼容版本](#)。

### 安裝 AWS ParallelCluster

#### 1. 用 pip 於安裝 AWS ParallelCluster。

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

當您使用 `--user` 參數時，pip 會安裝 AWS ParallelCluster 到 `~/.local/bin`。

2. 安裝節點版本管理員和最新的長期 Support (LTS) Node.js 版本。AWS Cloud Development Kit (AWS CDK) (AWS CDK) 需要 Node.js 才能產生 CloudFormation 範本。

#### Note

如果您的 Node.js 安裝無法在您的平台上運行，則可以在最新的 LTS 版本之前安裝 LTS 版本。如需詳細資訊，請參閱 [Node.js 版本排程](#) 和 [AWS CDK 先決條件](#)。

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

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

### 3. 確認已正確 AWS ParallelCluster 安裝。

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

### 4. 更新到最新版本的 & CLI，再次執行安裝命令。

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

## 安裝 AWS ParallelCluster 為獨立應用程式

在您的環境中安裝 AWS ParallelCluster 為獨立應用程式。請遵循下一節中在可用作業系統 AWS ParallelCluster 上安裝的指示。

### 必要條件

- 具有與可用安裝程式版本相容之作業系統的環境。

#### Note

AWS ParallelCluster 需要 NodeJS。AWS ParallelCluster 安裝程序包括 NodeJS ( v18 ) 的捆綁版本，如果它不存在，則會安裝該版本。如果您的系統與 NodeJS v18 不相容，您應該在安裝之前先安裝 NodeJS。AWS ParallelCluster

## Linux

### Linux x86 (64-bit)

安裝 AWS ParallelCluster 在您的環境中。

1. 下載最新的 [pcluster 安裝程式](#)。
2. 解壓縮安裝程式套裝軟體並使 AWS ParallelCluster 用下列指令進行安裝：

```
$ unzip pcluster-installer-bundle-3.10.1.713-node-v18.20.3-Linux_x86_64-signed.zip
-d pcluster-installer-bundle
$ cd pcluster-installer-bundle
$ chmod +x install_pcluster.sh
```

3. 執行下列安裝指令碼。

```
$ bash install_pcluster.sh
```

4. 確認已 AWS ParallelCluster 正確安裝。

```
$ pcluster version
{
  "version": "3.10.1"
}
```

### 排解pcluster安裝錯誤

- 如果在步驟 4 中未傳回 AWS ParallelCluster 版本，請重新啟動終端機或source更新PATH變數以包含新的二進位目錄，如下列範例所示：bash\_profile

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

- 如果您使用pcluster安裝建立具有CustomActions指定為 HTTPS 資源 (而非 S3 URI) 的叢集，您可能會看到一則WARNING訊息，指出這些資源可能未通過驗證 ([SSL: CERTIFICATE\_VERIFY\_FAILED])。這是由已知問題造成的，如果您信任指定資源的真實性，則可以忽略此警告。

### 先前的安裝程式套件

- 無

## 安裝後要採取的步驟

您可以通過運行驗證是否正確 AWS ParallelCluster 安裝 [pcluster version](#)。

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

AWS ParallelCluster 定期更新。若要更新至最新版本 AWS ParallelCluster，請再次執行安裝命令。如需有關最新版本的詳細資訊 AWS ParallelCluster，請參閱 [AWS ParallelCluster 版本說明](#)。

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

若要解除安裝 AWS ParallelCluster，請使用 `pip3 uninstall`。

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

如果您沒有 Python 和 pip3，請使用適用於您環境的程序。

## 安裝AWS ParallelCluster使用者介面

AWS ParallelClusterUI 是一個基於 Web 的用戶界面，可鏡像 AWS ParallelCluster pcluster CLI，同時提供類似控制台的體驗。您可以在AWS ParallelCluster您的AWS 帳戶。當您執行它時，AWS ParallelCluster使用者介面會存取 Amazon API Gateway 您的AWS 帳戶。AWS ParallelCluster 如需 AWS ParallelCluster UI 的詳細資訊，請參閱 [AWS ParallelCluster 介面](#)。

事前準備：

- 一個 AWS 帳戶
- 訪問 AWS Management Console

主題

- [安裝AWS ParallelCluster使用者介面](#)
- [建立自訂網域](#)
- [Amazon Cognito 使用者集區選項](#)
- [識別AWS ParallelCluster和 AWS ParallelCluster UI 版本](#)

- [將 AWS ParallelCluster UI 更新為新AWS ParallelCluster版本](#)
- [AWS ParallelClusterUI 成本](#)

## 安裝AWS ParallelCluster使用者介面

若要安裝 AWS ParallelCluster UI 的執行個體，您可以為在其中AWS CloudFormation建立叢集選擇快速建立連結。AWS 區域快速建立 URL 會帶您前往「建立堆疊精靈」，您可以在其中提供快速建立堆疊範本輸入並部署堆疊。如需 CloudFormation 快速建立堆疊的詳細資訊，請參閱《AWS CloudFormation使用指南》中的〈[建立堆疊的快速建立連結](#)〉。

### Note

您只能使用與安裝 AWS ParallelCluster UI 相同的AWS ParallelCluster版本來建立和編輯叢集或建置映像檔。

## AWS ParallelCluster用戶界面按地區快速創建鏈接

### UI 快速創建鏈接

[us-east-1](#)

[us-east-2](#)

[us-west-1](#)

[us-west-2](#)

[eu-west-1](#)

[eu-west-2](#)

[eu-west-3](#)

[eu-central-1](#)

[eu-north-1](#)

[me-south-1](#)

## UI 快速創建鏈接

[sa-east-1](#)

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

[ap-southeast-1](#)

[ap-southeast-2](#)

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

使用AWS CloudFormation快速建立連結部署具有巢狀 Amazon Cognito、API Gateway 和 Amazon EC2 Systems Manager 堆疊的 AWS ParallelCluster UI 堆疊。

1. 登入 AWS Management Console。
2. 從本節開頭的表格中選擇AWS 區域快速建立連結來部署 AWS ParallelCluster UI。這會帶您前往主控台內的 [CloudFormation 建立堆疊精靈]。
3. 輸入管理員電子郵件的有效電子郵件地址。

成功完成部署後，AWS ParallelClusterUI 會將臨時密碼傳送給您此電子郵件地址。您可以使用臨時密碼來訪問 AWS ParallelCluster UI。如果您在儲存或使用臨時密碼之前刪除電子郵件，則必須刪除堆疊並重新安裝 AWS ParallelCluster UI。

4. 將表單的其餘部分保持空白，或輸入 (可選) 參數的值以自訂 AWS ParallelCluster UI 組建。
5. 請記下堆疊名稱，以便在稍後的步驟中使用。
6. 導覽至 [權能]。同意功 CloudFormation 能。
7. 選擇建立。完成 AWS ParallelCluster API 和 AWS ParallelCluster UI 部署大約需要 15 分鐘。
8. 建立堆疊時檢視堆疊詳細資料。
9. 部署完成後，開啟傳送至您輸入之地址的管理員電子郵件。它包含用於存取 AWS ParallelCluster UI 的臨時密碼。如果您永久刪除電子郵件，但尚未登入 AWS ParallelCluster UI，則必須刪除您建立的 AWS ParallelCluster UI 堆疊，然後重新安裝 AWS ParallelCluster UI。



10. 在堆疊的AWS CloudFormation主控台清單中，選擇您在上一個步驟中記下的堆疊名稱的連結。
11. 在 [堆疊詳細資料] 中，選擇 [輸出]，然後選取名為 **Stackname** URL 的金鑰連結以開啟AWS ParallelCluster使用者介面。####是您在上一個步驟中記下的名稱。
12. 輸入臨時密碼。按照以下步驟創建自己的密碼並登錄。
13. 您現在位於所選 AWS ParallelCluster UI AWS 區域 的首頁上。
14. 若要開始使用 AWS ParallelCluster UI，請參閱[使 AWS ParallelCluster 用 UI 配置和創建集群](#)。

#### Note

PCUI 工作階段的預設持續時間為 5 分鐘，即 Cognito 在 PCUI 2023.12.0 時所提供的最小值。因此，預計從 Cognito 使用者集區移除的使用者仍然可以存取系統，直到工作階段過期為止。

## 建立自訂網域

了解如何為 AWS ParallelCluster UI 建立自訂網域。用戶界面託管在 Amazon API Gateway 在您的AWS帳戶。您可以在 API Gateway 主控台中建立自訂網域。

事前準備：

- 你有一個AWS帳戶。
- 您有可以存取的 AWS ParallelCluster UI 執行個體。
- 您擁有網域。
- 您可以變更基本的網域名稱系統 (DNS) 設定。

### 步驟 1：在 Amazon API Gateway 中建立新網域

1. 在中AWS Management Console，導覽至 [API Gateway](#)，您可以在其中看到列出的 AWS ParallelCluster UI API。
2. 在導覽窗格中，選擇 Custom domain names (自訂網域名稱)。
3. 選擇建立。
4. 在網域詳細資料中，輸入您的網域名稱。
5. 在 [端點組態] 中，選擇現有的 ACM 憑證，或選擇 [建立新的 ACM 憑證]。

(選擇性) 建立憑證

- a. 在 ACM 主控台中，選擇要求。
- b. 在網域名稱中，輸入您的網域名稱。
- c. 在驗證方法中，選擇驗證方法。

如果您選擇 [電子郵件驗證]，系統會將電子郵件傳送至網域註冊商登記的電子郵件地址。

- d. 選取 [我同意] 以啟動憑證。

## 步驟 2：設定 API 對應

1. 在 [[API Gateway](#)] 的 [自訂網域名稱] 中 your-domain-name，選擇 [設定 API 對應]。
2. 選擇 Custom domain names (自訂網域名稱)。
3. 選擇 Add new mapping (新增對應)。
4. 選擇 AWS ParallelCluster UI API、\$ 預設階段和儲存。
5. 在 API Gateway 網域名稱中，複製要在後續步驟中使用的值。

## 步驟三：設定 DNS 服務

- 建立 DNS CNAME 規則，將您的網域指向 API Gateway 網域。僅輸入網域。例如，請勿新增舞台，例如beta或prod。將 *abcde12345* 取代為您的 API Gateway 識別碼，並將###部 2 取代為 API。AWS 區域

規則	來源	目的地
CNAME	<i>example.com</i>	d-## 12345. ## API. ## ## -2. 亞馬遜

## 步驟 4：將網域新增至您的 Amazon Cognito 使用者集區

1. 導覽至 [Amazon Cognito 主控台](#)。
2. 選取您的使用者集區連結。
3. 選擇應用程式整合。
4. 在網域中，選擇動作 > 建立自訂網域。
5. 輸入您的自訂網域，然後選取您的 ACM 憑證。

6. 選擇「建立自訂網域」。

## 步驟 5：設定 API Gateway 回呼網址

1. 導覽至 [Amazon Cognito 主控台](#)。
2. 在 Amazon Cognito 使用者集區應用程式整合、應用程式用戶端和分析中，選取應用程式連結。
3. 在託管 UI 中，選擇編輯。
4. 在允許的回呼 URL 中，選擇新增其他 URL，然後輸入回呼 URL，例如example.com/login。

## 步驟 6：設定 Lambda 函數

1. 導覽至 [Lambda 主控台](#)。
2. 在導覽視窗中，選擇函數。
3. 篩選函數清單以尋找ParallelClusterUIFunction並選取連結。
4. 選擇配置，環境變量。
5. 選擇編輯。
6. 對於SITE\_URL值，請輸入您的自訂網域。
7. 導航到您的域，例如example.com，並進行身份驗證以連接到 AWS ParallelCluster UI。

## Amazon Cognito 使用者集區選項

以下各節介紹了 CloudFormation 快速創建鏈接或快速創建 URL。快速建立 URL 會帶您前往「建立堆疊精靈」，您可以在其中提供快速建立堆疊範本輸入並部署堆疊。如需 CloudFormation 快速建立堆疊的詳細資訊，請參閱《AWS CloudFormation使用指南》中的 [〈建立堆疊的快速建立連結〉](#)。

若要維護可用於多個 AWS ParallelCluster UI 執行個體的 Amazon Cognito 使用者集區，請考慮下列選項：

- 使用現有的 AWS ParallelCluster UI 執行個體，該執行個體可連結至從巢狀 CloudFormation 堆疊建立的 Amazon Cognito 使用者集區。這是您使用快速建立連結部署 AWS ParallelCluster UI 時所建立的，並將所有 Amazon Cognito 參數保持空白。
- 使用在部署 AWS ParallelCluster UI 之前部署的獨立 Amazon Cognito 使用者集區。然後，部署連結至您已部署的獨立 Amazon Cognito 使用者集區的新 AWS ParallelCluster UI 執行個體。如此一來，您就可以將 Amazon Cognito 部署與AWS ParallelCluster使用者介面部署分開來。此外，非巢狀 AWS ParallelCluster UI CloudFormation 堆疊更容易更新。

## 將現有的 Amazon Cognito 使用者集區與新的 AWS ParallelCluster UI 執行個體搭配使用

1. 在CloudFormation 主控台中，選取包含要用於多個 AWS ParallelCluster UI 執行個體之 Amazon Cognito 使用者集區的 AWS ParallelCluster UI 堆疊。
2. 瀏覽至建立 Amazon Cognito 使用者集區的巢狀堆疊。
3. 選取 Outputs (輸出) 標籤。
4. 複製下列參數的值：
  - UserPoolId
  - UserPoolAuthDomain
  - SNSRole
5. 使用快速建立連結部署新的 AWS ParallelCluster UI 執行個體，並使用複製的輸出填入所有External AWS ParallelCluster UI Amazon Cognito參數。這樣可防止新的 AWS ParallelCluster UI 堆疊建立新集區，並將其連結至從巢狀堆疊建立的現有 Amazon Cognito 使用者集區。您可以部署具有相同參數值的後續新 AWS ParallelCluster UI 執行個體，也可以將它們連結至 Amazon Cognito 使用者集區。

## 創建一個獨立 Amazon Cognito 戶池

AWS ParallelCluster用戶界面 Amazon Cognito 按區域快速創建鏈接

用戶界面 Amazon Cognito 快速創建鏈接

[us-east-1](#)

[us-east-2](#)

[us-west-1](#)

[us-west-2](#)

[eu-west-1](#)

[eu-west-2](#)

[eu-west-3](#)

## 用戶界面 Amazon Cognito 快速創建鏈接

[eu-central-1](#)

[eu-north-1](#)

[me-south-1](#)

[sa-east-1](#)

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

[ap-southeast-1](#)

[ap-southeast-2](#)

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

1. 選擇一個標記為與您部署 UI 執行個體相同的快速建立連結，啟動僅限 Amazon Cognito AWS 區域的堆疊。AWS ParallelCluster請參閱本節開頭的快速建立連結。
2. 堆疊建立完成後，選取「輸出」頁籤並複製下列參數的值：
  - UserPoolId
  - UserPoolAuthDomain
  - SNSRole
3. 選擇 AWS ParallelCluster AWS ParallelCluster UI 快速入門連結，並使用您複製的值填入所有 External AWS ParallelCluster UI Amazon Cognito 參數，以部署新的 UI 執行個體。新的 AWS ParallelCluster UI 執行個體會連結到獨立的 Amazon Cognito 使用者集區，而不會建立巢狀堆疊或新的使用者集區。您可以部署具有相同參數值的後續新 AWS ParallelCluster UI 執行個體，也可以將它們連結到獨立的 Amazon Cognito 使用者集區。

## 識別AWS ParallelCluster和 AWS ParallelCluster UI 版本

1. 在主 CloudFormation 控台中，選取 AWS ParallelCluster UI 堆疊。
2. 選取「參數」頁籤。
3. AWS ParallelCluster版本是參數版本的值。
4. AWS ParallelClusterUI 版本位於PublicEcrImageUri值的末尾。例如，如果值為public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02，則版本為2023.02。

## 將 AWS ParallelCluster UI 更新為新AWS ParallelCluster版本

若要將 AWS ParallelCluster UI 更新為最新AWS ParallelCluster版本，請選擇[快速建立連結](#)來啟動新堆疊。

## AWS ParallelClusterUI 成本

AWS ParallelClusterUI 建立在無伺服器架構上，在大多數情況下，您可以在AWS免費方案類別中使用它。下表列出 AWS ParallelCluster UI 所依賴AWS 服務的項目及其免費方案限制。典型用量估計每個月的成本不到一美元。

服務	AWS 免費方案
Amazon Cognito	每月 5 萬個活躍用戶
Amazon API Gateway	1 百萬個休息 API 調用
AWS Lambda	每月 100 萬個免費請求，每月 40 萬 GB 的運算時間
EC2 Image Builder	無需任何費用，EC2 除外
Amazon Elastic Compute Cloud	15 分鐘的一次性容器映像
AWS CloudFormation	5 GB 資料 (擷取、封存儲存和日誌見解查詢掃描的資料)

# 開始使用 AWS ParallelCluster

透過使用 AWS ParallelCluster 命令列介面 (CLI) 或網頁式使用者介面 (UI) 來開始設定和建立叢集。AWS ParallelCluster 使用者介面已在 3.5.0 版中新增。

## 主題

- [使用 AWS ParallelCluster 命令行界面配置和創建集群](#)
- [使 AWS ParallelCluster 用 UI 配置和創建集群](#)
- [連接至叢集](#)

## 使用 AWS ParallelCluster 命令行界面配置和創建集群

安裝之後 AWS ParallelCluster，請完成下列設定步驟。

確認您的 AWS 帳戶具有包含執行 [pcluster](#) CLI 所需權限的角色。如需詳細資訊，請參閱 [AWS ParallelCluster pcluster 使用者策略範例](#)。

設定您的 AWS 認證。如需詳細資訊，請參閱 [使用](#) 指南 AWS CLI 中的 AWS CLI 〈配置〉。

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

啟動叢集的 AWS 區域 位置必須至少有一個 Amazon EC2 key pair。如需詳細資訊，請參閱 [Amazon 彈性運算雲端 Linux 執行個體使用者指南](#) 中的 [Amazon 彈性運算雲端金鑰對](#)。

使用命 AWS ParallelCluster 令列介面 (CLI) 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 使用的服務 AWS ParallelCluster](#)。

## 設定並建立您的第一個叢集

使用 `pcluster configure` CLI 命令初始化精靈，提示您輸入設定和建立叢集所需的所有資訊，以建立您的第一個叢集。與使用相比，AWS Batch 作為調度程序使用時，序列的詳細信息不同 Slurm。

## Slurm

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

從有效 AWS 區域 識別碼清單中，選擇 AWS 區域 您希望叢集執行的位置。

 Note

AWS 區域 顯示的列表基於您的帳戶的分區，並且僅包括為您 AWS 區域 的帳戶啟用的分區。如需有關為您的帳戶啟 AWS 區域 用的詳細資訊，請參閱 AWS 區域中的[管理AWS 一般參考](#)。顯示的範例來自 AWS 全域磁碟分割。如果您的帳戶位於 AWS GovCloud (US) 分區 AWS 區域 中，則僅列出 ( gov-us-east-1和gov-us-west-1 ) 該分區中。同樣，如果您的帳戶位於中 AWS 國分區中，則僅顯示cn-north-1並cn-northwest-1顯示。如需 AWS 區域 支援的完整清單 AWS ParallelCluster，請參閱[AWS ParallelCluster 支援的區域](#)。

Allowed values for 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]:

key pair 是從所選取的 Amazon 彈性運算雲端註冊的金鑰配對中選取的 AWS 區域。選擇 key pair :

Allowed values for Amazon EC2 Key Pair Name:

1. your-key-1



```
2. your-key-2
Amazon EC2 Key Pair Name [your-key-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. ubuntu2204
4. ubuntu2004
5. rhel8
Operating System [alinux2]:
```

選擇頭節點實例類型：

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

選擇佇列組態。注意：無法為同一佇列中的多個計算資源指定執行個體類型。

```
Number of queues [1]:
Name of queue 1 [queue1]:
Number of compute resources for queue1 [1]: 2
Compute instance type for compute resource 1 in queue1 [t2.micro]:
Maximum instance count [10]:
```

讓 EFA 執行需要大規模執行個體間通訊的應用程式，無需額外付費：AWS

- 選擇[支援 Elastic Fabric Adapter \(EFA\)](#) 的執行個體類型。
- 啟用[全民福利協會](#)。
- 指定現有的[放置群組](#)名稱。如果將其保留空白，請為您 AWS ParallelCluster 創建一個。

```
Compute instance type for compute resource 2 in queue1 [t2.micro]: c5n.18xlarge
```

```
Enable EFA on c5n.18xlarge (y/n) [y]: y
Maximum instance count [10]:
Placement Group name []:
```

完成上述步驟後，決定是使用現有的 VPC 還是讓您 AWS ParallelCluster 建立 VPC。如果您沒有正確設定的 VPC，AWS ParallelCluster 可以為您建立新的 VPC。它可以將頭節點和計算節點放在相同的公有子網路中，或者只將頭節點放在公有子網路中，其中包含所有運算節點位於私有子網路中。如果您允許 AWS ParallelCluster 建立 VPC，則必須決定是否所有節點都位於公用子網路中。如需詳細資訊，請參閱 [網路組態](#)。

如果您將叢集設定為使用具有多個網路介面或網路卡的執行個體類型，請參閱 [網路組態](#) 以瞭解其他網路需求。

有可能達到您的配額，以達到 AWS 區域。預設配額為五個 VPC (用於 AWS 區域)。如需有關此配額以及如何申請增加的詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC [和子網路](#)。

#### Important

依預設，由建立的 VPC AWS ParallelCluster 不會啟用 VPC 流程記錄。VPC 流程記錄可讓您擷取 VPC 中進出網路介面的 IP 流量相關資訊。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [VPC 流量日誌](#)。

如果您允許 AWS ParallelCluster 建立 VPC，請確定您決定是否所有節點都位於公用子網路中。

#### Note

如果您選擇 1. Head node in a public subnet and compute fleet in a private subnet，即使您指定了免費方案資源，也 AWS ParallelCluster 會建立 NAT 閘道，以產生額外費用。

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
```

```

Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized

```

如果不建立新的 VPC，則必須選取現有的 VPC。

如果您選擇 AWS ParallelCluster 建立 VPC，請記下 VPC ID，以便稍後使用 AWS CLI 將其刪除。

```

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 VPC 之後，決定是否要使用現有的子網路或是建立新的子網路。

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

```

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

```

## AWS Batch

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

從有效 AWS 區域 識別碼清單中，選擇 AWS 區域 您希望叢集執行的位置。

### Note

AWS 區域 顯示的列表是根據您帳戶的分區而定。它僅包括 AWS 區域 為您的帳戶啟用的功能。如需有關為您的帳戶啟 AWS 區域 用的詳細資訊，請參閱 AWS 區域中的[管理AWS 一般參考](#)。顯示的範例來自 AWS 全域磁碟分割。如果您的帳戶位於 AWS GovCloud (US) 分區 AWS 區域 中，則僅列出 ( gov-us-east-1和gov-us-west-1 ) 該分區中。同樣，如

果您的帳戶位於中 AWS 國分區中，則僅顯示cn-north-1並cn-northwest-1顯示。如需 AWS 區域 支援的完整清單 AWS ParallelCluster，請參閱[AWS ParallelCluster 支援的區域](#)。

Allowed values for 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 [us-east-1]:

金鑰對是從所選取的 Amazon EC2 註冊的 key pair 中選取的 AWS 區域。選擇 key pair :

Allowed values for Amazon EC2 Key Pair Name:

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

Amazon EC2 Key Pair Name [your-key-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]: 2

當選取 `awsbatch` 做為排程器時，`alinux2` 會用作為作業系統。輸入的頭節點實例類型如下：

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

選擇佇列組態。AWS Batch 排程器只包含單一佇列。輸入計算節點叢集的大小上限。這是以 vCPU 為單位測量而得。

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

決定要使用現有的 VPC 還是讓您 AWS ParallelCluster 建立 VPC。如果您沒有適當設定的 VPC，AWS ParallelCluster 可以建立一個新的 VPC。它可以同時使用相同公有子網路中的頭節點和計算節點，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。您可以根據區域允許的 VPC 數量達到配額。VPC 的預設數目為五。如需有關此配額以及如何申請增加的詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC [和子網路](#)。

#### Important

依預設，由建立的 VPC AWS ParallelCluster 不會啟用 VPC 流程記錄。VPC 流程記錄可讓您擷取 VPC 中進出網路介面的 IP 流量相關資訊。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [VPC 流量日誌](#)。

如果您允許 AWS ParallelCluster 建立 VPC，請確定您決定是否所有節點都位於公用子網路中。

#### Note

如果您選擇 1. Head node in a public subnet and compute fleet in a private subnet，即使您指定了免費方案資源，也 AWS ParallelCluster 會建立 NAT 閘道，以產生額外費用。

```
Automate VPC creation? (y/n) [n]: y  
Allowed values for Availability Zone:  
1. us-east-1a  
2. us-east-1b  
3. us-east-1c  
4. us-east-1d
```

```

5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
  subnet]: *1*
Beginning VPC creation. Please do not leave the terminal until the creation is
  finalized

```

如果不建立新的 VPC，則必須選取現有的 VPC。

如果您選擇 AWS ParallelCluster 建立 VPC，請記下 VPC ID，以便稍後使用 AWS CLI 或刪 AWS Management Console 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 VPC 之後，請確定您決定要使用現有子網路還是建立新子網路。

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

```

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

```

完成上述步驟後，一個簡單的叢集會啟動到 VPC 中。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：

- VPC 必須具有 DNS Resolution = yes 和 DNS Hostnames = yes。
- VPC 端也必須具有正確 domain-name 的 DHCP 選項。AWS 區域預設 DHCP 選項集已指定必要 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 來啟用運算節點的 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-cluster --cluster-name test-cluster --cluster-configuration cluster-  
config.yaml  
{  
  "cluster": {  
    "clusterName": "test-cluster",  
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",  
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/test-cluster/  
abcdef0-f678-890a-5abc-021345abcdef",  
    "region": "eu-west-1",  
    "version": "3.7.0",  
    "clusterStatus": "CREATE_IN_PROGRESS"  
  },  
  "validationMessages": []  
}
```

追蹤叢集進度：

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

或

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

叢集達到"clusterStatus": "CREATE\_COMPLETE"狀態後，您可以使用一般 SSH 用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 Amazon EC2 使用者指南。或者您可以通過以下方式連接集群

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

若要刪除叢集，請執行下列命令。

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

刪除叢集之後，您可以透過刪除網路堆疊來刪除 VPC 中的 CloudFormation 網路資源。堆疊的名稱以「平行叢集網路」開頭，並以「YYYYMMDDHMMSS」格式包含建立時間。您可以使用 [list-stacks](#) 指令列出堆疊。

```
$ aws --region us-east-1 cloudformation list-stacks \
```

```
--stack-status-filter "CREATE_COMPLETE" \  
--query "StackSummaries[].StackName" | \  
grep -e "parallelclusternetworking-" \  
"parallelclusternetworking-pubpriv-20191029205804"
```

可以使用 [delete-stack](#) 指令刪除堆疊。

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

為您建 [pcluster configure](#) 立的 VPC 不會在 CloudFormation 網路堆疊中建立。您可以在主控台中手動刪除該 VPC，或使用 AWS CLI。

```
$ aws --region us-east-1 Amazon EC2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

## 使 AWS ParallelCluster 用 UI 配置和創建集群

AWS ParallelCluster UI 是一個基於 Web 的用戶界面，可鏡像 AWS ParallelCluster pcluster CLI，同時提供類似控制台的體驗。您可以在 AWS ParallelCluster 您的 AWS 帳戶。當您執行它時，AWS ParallelCluster 使用者介面會存取 Amazon API Gateway 您的 AWS 帳戶。AWS ParallelCluster

### Note

AWS ParallelCluster UI 精靈可能沒有最新支援 AWS ParallelCluster 版本中所有支援功能的 UI 選項。您可以視需要手動編輯組態檔或使用 AWS ParallelCluster CLI。

在本節中，我們將引導您使用 AWS ParallelCluster UI 配置和創建叢集。

事前準備：

- 存取執行中的 AWS ParallelCluster UI 執行個體。如需詳細資訊，請參閱 [安裝AWS ParallelCluster 使用者介面](#)。

### 配置和建立叢集

- 在 AWS ParallelCluster UI 叢集檢視中，選擇建立叢集，逐步。
- 在叢集的名稱中，輸入叢集的名稱。



3. 為您的叢集選擇具有公用子網路的 VPC，然後選擇下一步。
4. 在 [頭] 節點中，選擇 [新增 SSM 工作階段]，然後選擇 [下一步]
5. 在 [佇列] 的 [計算資源] 中，針對靜態節點選擇 1。
6. 對於「執行個體類型」，移除選取的預設執行個體類型，選擇 t2.micro，然後選擇「下一步」。
7. 在儲存空間中，選擇下一步。
8. 在叢集配置中，檢閱叢集配置 YAML，然後選擇乾式執行以驗證它。
9. 選擇建立以根據驗證的組態建立叢集。
10. 幾秒鐘後，AWS ParallelCluster UI 會自動將您導覽回叢集，您可以在其中監視叢集建立狀態和 Stack 事件。
11. 選擇 [詳細資料] 以查看叢集詳細資料，例如版本和狀態。
12. 選擇執行個體以查看 Amazon EC2 執行個體和狀態的清單。
13. 選擇堆疊事件以檢視叢集堆疊事件，以及建立叢集之 CloudFormation 堆疊的 AWS Management Console 連結。
14. 在詳細資料中，叢集建立完成之後，選擇檢視 YAML 以檢視或下載叢集組態 YAML 檔案。
15. 叢集建立完成後，選擇命令介面以存取叢集標頭節點。

#### Note

當您選擇命令介面時，會 AWS ParallelCluster 開啟 Amazon EC2 Systems Manager 工作階段並新增 `ssm-user` 至 `/etc/sudoers`。如需詳細資訊，請參閱 Amazon EC2 系統管理員使用指南中的開啟或關閉 `ssm-user` 帳戶管理許可。

16. 若要清理，請在 [叢集] 檢視中選取叢集，然後選擇動作，刪除叢集。

## 連接至叢集

使用時 AWS ParallelCluster，您可以連線至叢集標頭節點以執行工作、檢視結果、管理使用者，以及監視叢集和工作狀態。使用下列方法 Connect 至叢集標頭節點執行個體：

- 使用 `ssh` [key pair 組](#) 登入。在叢集配置中指定 [HeadNode/KeyName](#) 中的私密金鑰。如需詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的 [使用 SSH 連接至 Linux 執行個體](#)。
- 使用 `pcluster ssh` 指令行介面 (CLI) 命令登入。在叢集配置 [HeadNode/](#) 中指定私密金鑰 [KeyName](#)。如需詳細資訊，請參閱 [pcluster ssh](#)。

- 使用 SSM 工作階段 Connect 至叢集標頭節點。您必須將AmazonSSMManagedInstanceCore受管理的原則新增至叢集配置[AdditionalIamPolicies](#)中的 [HeadNode/](#)，才能使用 SSM 工作階段進行連線。如需詳細資訊，請參閱《[SSM 使用者指南](#)》中的《[SSM 工作階段管理員](#)》。
- 使用 NICE DCV Connect 至叢集頭節點。如需詳細資訊，請參閱 [通過 NICE DCV Connect 到頭節點](#)。
- 使用 AWS ParallelCluster UI 時，您也可以使用 UI 提供的 Amazon EC2 Connect 命令來連接到叢集頭節點。

## 多個使用者存取叢集

瞭解如何實作和管理單一叢集的多個使用者存取權。

在本主題中，AWS ParallelCluster 使用者是指運算執行個體的系統使用者。一個例子是ec2-user一個 Amazon EC2 實例。

AWS ParallelCluster 多用戶訪問支持在所有當前可用 AWS ParallelCluster 的 AWS 區域 地方。它適用於其他 AWS 服務，包括 [Amazon FSx for Lustre](#) 和 [Amazon Elastic File System](#)。

您可以使用[AWS Directory Service for Microsoft Active Directory](#)或 [Simple AD](#) 來管理叢集存取。請務必檢查這些服務的[AWS 區域 可用性](#)。若要設定叢集，請指定組[AWS ParallelCluster DirectoryService](#)態。AWS Directory Service 目錄可以連接到多個叢集。這可讓您集中管理多個環境的身分識別，並提供統一的登入體驗。

當您用 AWS Directory Service 於 AWS ParallelCluster 多個使用者存取時，您可以使用目錄中定義的使用者認證登入叢集。這些認證包含使用者名稱和密碼。第一次登入叢集之後，系統會自動產生使用者安全殼層金鑰。您可以使用它在沒有密碼的情況下登錄。

您可以在部署目錄服務之後建立、刪除和修改叢集的使用者或群組。使用 AWS Directory Service，您可以在 AWS Management Console 或使用 Active Directory 使用者和電腦工具中執行此操作。您可以從任何已加入您活動目錄的 Amazon EC2 執行個體存取此工具。如需詳細資訊，請參閱[安裝 Active Directory 管理工具](#)。

如果您打算 AWS ParallelCluster 在沒有網際網路存取權的單一子網路中使用，請參閱[AWS ParallelCluster 在沒有互聯網訪問的單個子網中](#)以取得其他需求。

### 主題

- [創建一個活動目錄](#)
- [建立具有 AD 網域的叢集](#)

- [登入與 AD 網域整合的叢集](#)
- [執行 MPI 工作](#)
- [AWS Managed Microsoft AD 透過 LDAP \(S\) 叢集配置的範例](#)

## 創建一個活動目錄

在建立叢集之前，請確定您已建立作用中目錄 (AD)。如需有關如何為叢集選擇 Active Directory 類型的詳細資訊，請參閱《AWS Directory Service 管理指南》中的[選擇](#)項目。

如果目錄為空，請使用使用者名稱和密碼新增使用者。如需詳細資訊，請參閱 [Simple AD](#) 的 [AWS Directory Service for Microsoft Active Directory](#) 特定文件。

### Note

AWS ParallelCluster 要求每個使用中目錄使用者目錄都位於 `/home/$user` 目錄中。

## 建立具有 AD 網域的叢集

### Warning

本簡介章節說明如何透過輕量型目錄存取通訊協定 (LDAP) 使用受管理的 Active Directory (AD) 伺服器進行設定。AWS ParallelCluster LDAP 是一種不安全的通訊協定。對於生產系統，我們強烈建議使用 TLS 憑證 (LDAPS)，如下一[AWS Managed Microsoft AD 透過 LDAP \(S\) 叢集配置的範例](#)節所述。

透過在叢集配置檔 `DirectoryService` 區段中指定相關資訊，將叢集設定為與目錄整合。如需詳細資訊，請參閱 [DirectoryService](#) 組態一節。

您可以使用下列範例將叢集與 AWS Managed Microsoft AD 透過輕量型目錄存取通訊協定 (LDAP) 整合。

AWS Managed Microsoft AD 透過 LDAP 組態所需的特定定義：

- 您必須 `True` 在 [DirectoryService](#) 之下將 `ldap_auth_disable_tls_never_use_in_production` 參數設定為 [AdditionalSssdConfigs](#)。

- 您可以為 [DirectoryService/DomainAddr](#) 指定控制器主機名稱或 IP 位址。
- [DirectoryService/DomainReadOnlyUser](#) 語法必須如下：

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

取得您的 AWS Managed Microsoft AD 組態資料：

```
$ aws ds describe-directories --directory-id "d-abcdef01234567890"
```

```
{
  "DirectoryDescriptions": [
    {
      "DirectoryId": "d-abcdef01234567890",
      "Name": "corp.example.com",
      "DnsIpAddrs": [
        "203.0.113.225",
        "192.0.2.254"
      ],
      "VpcSettings": {
        "VpcId": "vpc-021345abcdef6789",
        "SubnetIds": [
          "subnet-1234567890abcdef0",
          "subnet-abcdef01234567890"
        ],
        "AvailabilityZones": [
          "region-idb",
          "region-idd"
        ]
      }
    }
  ]
}
```

以 AWS Managed Microsoft AD 下項目的叢集配置：

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
```

```
Networking:
  SubnetId: subnet-1234567890abcdef0
Ssh:
  KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

若要針對 Simple AD 使用此組態，請變更 **DirectoryService** 區段中的 **DomainReadOnlyUser** 屬性值：

```
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:SimpleAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

考量：

- 建議您透過 TLS/SSL (或 LDAPS) 使用 LDAP，而不是單獨使用 LDAP。TLS/SSL 可確保連線已加密。

- [DirectoryService/DomainAddr](#) 屬性值與 `describe-directories` 輸出 `DnsIpAddrs` 清單中的項目相符。
- 建議您的叢集使用位於 [DirectoryService/DomainAddr](#) 指向的相同可用區域中的子網路。如果您使用針對目錄 VPC 建議的 [自訂動態主機設定通訊協定 \(DHCP\) 組態](#)，且您的子網路不在 [DirectoryService/DomainAddr](#) 可用區域中，則可能會在可用區域之間進行交叉流量。使用多使用者 AD 整合功能不需要使用自訂 DHCP 設定。
- [DirectoryService/DomainReadOnlyUser](#) 屬性值指定必須在目錄中建立的使用者。依預設，不會建立此使用者。建議您不要授與此使用者修改目錄資料的權限。
- [DirectoryService/PasswordSecretArn](#) 屬性值指向一個 AWS Secrets Manager 密碼，其中包含您為 [DirectoryService/DomainReadOnlyUser](#) 屬性指定的使用者密碼。如果此使用者的密碼變更，請更新密碼值並更新叢集。若要更新新密碼值的叢集，您必須使用 `pcluster update-compute-fleet` 指令停止計算叢集。如果您已將叢集設定為使用 [LoginNodes](#)，請在將 [LoginNodes//設定Count](#) 為 0 之後停止 [LoginNodesPools/Pools](#) 並更新叢集。然後，從叢集頭節點內執行下列命令。

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

如需其他範例，另請參閱 [整合作用中目錄](#)。

## 登入與 AD 網域整合的叢集

如果您啟用了主動傳遞 (AD) 網域整合功能，則會在叢集頭節點上啟用密碼驗證。AD 使用者的主目錄是在第一次使用者登入頭節點時建立，或是在頭節點上第一次 `sudo-user` 切換至 AD 使用者時建立。

叢集運算節點未啟用密碼驗證。AD 使用者必須登入才能使用安全殼層金鑰計算節點。

根據預設，SSH 金鑰會在首次登入頭節點時，在 AD 使用者/`/${HOME}/.ssh` 目錄中設定。您可以 `false` 在叢集配置中將 [DirectoryService/GenerateSshKeysForUsers](#) `boolean` 屬性設定為來停用此行為。默認情況下 [GenerateSshKeysForUsers](#)，[DirectoryService/](#) 設置為 `true`。

如果 AWS ParallelCluster 應用程式在叢集節點之間需要無密碼 SSH，請確定 SSH 金鑰已在使用者的主目錄中正確設定。

AWS Managed Microsoft AD 密碼在 42 天後過期。如需詳細資訊，請參閱 [《管理指南》AWS Managed Microsoft AD 中的 AWS Directory Service](#) 〈管理密碼原則〉。如果您的密碼過期，則必須重設密碼才能恢復叢集存取。如需詳細資訊，請參閱 [如何重置用戶密碼和過期的密碼](#)。

**Note**

如果 AD 整合功能無法如預期般運作，SSSD 記錄檔可提供有用的診斷資訊，以便疑難排解問題。這些記錄檔位於叢集節點上的 `/var/log/sss` 目錄中。依預設，它們也會儲存在叢集的 Amazon CloudWatch 日誌群組中。

如需詳細資訊，請參閱 [疑難排解多使用者與活動目錄整合](#)。

## 執行 MPI 工作

如 SchedMD 所建議，啟動 MPI 工作使用 Slurm 做為 MPI 啟動載入方法。[有關更多信息，請參閱您的 MPI 圖書館的官方文檔或官方文檔。Slurm](#)

例如，在 [IntelMPI 官方文件](#) 中，您會學到執行 StarCCM 工作時，您必須透過匯出環境變數來設定 Slurm 為程序協調器。`I_MPI_HYDRA_BOOTSTRAP=slurm`

**Note****已知問題**

如果您的 MPI 應用程式依賴 SSH 作為機制來產生 MPI 工作，則可能會發生 [Slurm 中的已知錯誤](#)，導致錯誤的目錄使用者名稱解析為「無人」。

您可以將應用程式設定 Slurm 為使用 MPI 啟動載入方法，或參閱疑難排解一節 [用戶名解析的已知問題](#) 中的詳細資訊和可能的因應措施。

## AWS Managed Microsoft AD 透過 LDAP (S) 叢集配置的範例

AWS ParallelCluster 透過輕量型目錄存取通訊協定 (LDAP) 或透 AWS Directory Service 過 TLS/SSL (LDAPS) 的 LDAP 整合，支援多重使用者存取。

下列範例顯示如何建立叢集配置以與 AWS Managed Microsoft AD 透過 LDAP (S) 整合。

AWS Managed Microsoft AD 超過 LDAPS，具有憑證驗證

您可以使用此範例，透過憑證驗證來整合叢集與 AWS Managed Microsoft AD 超過 LDAPS。

具有憑證組態的 AWS Managed Microsoft AD 超過 LDAPS 的特定定義：

- [DirectoryService](#) 對於具有憑證驗證的 LDAPS，[LdapTlsReqCert](#) 必須將/設定為 hard (預設)。

- [DirectoryService/LdapTlsCaCert](#) 必須指定授權憑證 (CA) 憑證的路徑。

CA 憑證是一種憑證服務包，其中包含針對 AD 網域控制站發行憑證的整個 CA 鏈結的憑證。

您的 CA 憑證和憑證必須安裝在叢集節點上。

- 必須為 [DirectoryService](#) 指定控制器主機名稱 [DomainAddr](#)，而不是 IP 位址。
- [DirectoryService/DomainReadOnlyUser](#) 語法必須如下：

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

透過 LDAPS 使用 AD 的叢集配置檔案範例：

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
  CustomActions:
    OnNodeConfigured:
      Script: s3://aws-parallelcluster/scripts/pcluster-dub-msad-ldaps.post.sh
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
      Iam:
        AdditionalIamPolicies:
```



```

    - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
  CustomActions:
    OnNodeConfigured:
      Script: s3://aws-parallelcluster-pcluster/scripts/pcluster-dub-msad-
ldaps.post.sh
  DirectoryService:
    DomainName: dc=corp,dc=example,dc=com
    DomainAddr: ldaps://win-abcdef01234567890.corp.example.com,ldaps://win-
abcdef01234567890.corp.example.com
    PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
    DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
    LdapTlsCaCert: /etc/openldap/cacerts/corp.example.com.bundleca.cer
    LdapTlsReqCert: hard

```

在安裝後指令碼中新增憑證並設定網域控制站：

```

*#!/bin/bash*
set -e

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

AD_HOSTNAME_1="win-abcdef01234567890.corp.example.com"
AD_IP_1="192.0.2.254"

AD_HOSTNAME_2="win-abcdef01234567890.corp.example.com"
AD_IP_2="203.0.113.225"

# Download CA certificate
mkdir -p $(dirname "${AD_CERTIFICATE_LOCAL}")
aws s3 cp "${AD_CERTIFICATE_S3_URI}" "${AD_CERTIFICATE_LOCAL}"
chmod 644 "${AD_CERTIFICATE_LOCAL}"

# Configure domain controllers reachability
echo "${AD_IP_1} ${AD_HOSTNAME_1}" >> /etc/hosts
echo "${AD_IP_2} ${AD_HOSTNAME_2}" >> /etc/hosts

```

您可以從加入網域的執行個體擷取網域控制站主機名稱，如下列範例所示。

從視窗執行個體

```
$ nslookup 192.0.2.254
```

```
Server: corp.example.com
Address: 192.0.2.254

Name: win-abcdef01234567890.corp.example.com
Address: 192.0.2.254
```

來自 Linux 執行個體

```
$ nslookup 192.0.2.254
```

```
192.0.2.254.in-addr.arpa name = corp.example.com
192.0.2.254.in-addr.arpa name = win-abcdef01234567890.corp.example.com
```

AWS Managed Microsoft AD 超過 LDAPS，無需驗證憑證

您可以使用此範例將叢集與 AWS Managed Microsoft AD 超過 LDAPS 整合，而無需驗證憑證。

不使用憑證驗證組態的 AWS Managed Microsoft AD 超過 LDAPS 的特定定義：

- [DirectoryService/LdapTlsReqCert](#) 必須設定為 `never`。
- 您可以為 [DirectoryService/DomainAddr](#) 指定控制器主機名稱或 IP 位址。
- [DirectoryService/DomainReadOnlyUser](#) 語法必須如下：

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

在沒有憑證驗證的情況下 AWS Managed Microsoft AD 透過 LDAPS 使用的叢集配置檔範例：

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
```

```
- Name: queue1
  ComputeResources:
    - Name: t2micro
      InstanceType: t2.micro
      MinCount: 1
      MaxCount: 10
  Networking:
    SubnetIds:
      - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://203.0.113.225,ldaps://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsReqCert: never
```

## 最佳實務

### 最佳實踐：頭節點實例類型選擇

即使頭節點不執行作業，其功能及其大小對叢集的整體效能至關重要。當您選擇要用於主節點的執行個體類型時，請考慮下列特性：

**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。若要擴展和縮減具有大量節點的叢集，請為主節點提供一些額外的運算容量。

**共用檔案系統：**使用共用檔案系統時，請選擇具有足夠網路頻寬和足夠 Amazon EBS 頻寬的執行個體類型來處理工作流程。確定頭節點能夠為叢集公開足夠的 NFS 伺服器目錄，並處理需要在運算節點和主節點之間共用的成品。

### 最佳做法：網路效能

網路效能對於高效能運算 (HPC) 應用程式而言至關重要。如果沒有可靠的網路效能，這些應用程式無法如預期般執行。若要最佳化網路效能，請考慮下列最佳作法。

- **放置群組：**如果您正在使用 Slurm，請考慮將每個 Slurm 佇列設定為使用叢集置放群組。叢集的置放群組是單一可用區域內執行個體的邏輯群組。如需詳細資訊，請參閱 [Amazon EC2 使用者指南中的放置群組](#)。您可以在佇列的 [Networking](#) 區段 [PlacementGroup](#) 中指定一個，每個計算資源都會指派給佇列的放置群組。在計算資源的 [Networking](#) 區段 [PlacementGroup](#) 中指定

時，會將該特定計算資源指派給該放置群組。計算資源放置群組規格會覆寫計算資源的佇列規格。如需詳細資訊，請參閱 [SlurmQueuesNetworking/SlurmQueues/PlacementGroup](#) 和 [ComputeResources/Networking/PlacementGroup](#)。

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

或者，也可以為您 AWS ParallelCluster 建立放置群組。

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

從 3.3.0 AWS ParallelCluster 版開始，會修改置放群組的建立和管理。當您指定要啟用的放置群組時，佇列中沒有name或Id，每個計算資源都會指派自己的受管理放置群組，而不是為整個佇列指派一個受管理群組。這有助於減少容量不足的錯誤。如果整個佇列需要有一個放置群組，您可以使用具名放置群組。

[SlurmQueues/Networking/PlacementGroup/Name](#)已加入為//[SlurmQueuesNetworkingPlacementGroup/](#)的偏好替代方案Id。

如需詳細資訊，請參閱 [Networking](#)。

- 增強型聯網：請考慮選擇支援增強型聯網的執行個體類型。此建議適用於所有[目前一代的執行個體](#)。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與執行個體通訊，請考慮為您的網路選擇 EFA 網路介面。EFA 的自訂作業系統 (OS) 略過硬體可增強執行個體與執行個體之間的通訊，並具有的隨需彈性和彈性。AWS 雲端您可以配置[ComputeResource](#)要使用的每個Slurm佇列Efa。如需搭配使用 EFA 的詳細資訊 AWS ParallelCluster，請參閱[Elastic Fabric Adapter](#)。

```
ComputeResources:
  - Name: your-compute-resource-name
    Efa:
      Enabled: true
```

如需有關 EFA 的詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的[彈性網狀架構配接器](#)。

- 執行個體頻寬：頻寬會隨執行個體大小調整。如需有關不同執行個體類型的資訊，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

若要管理中的資源成本 AWS ParallelCluster，建議您使用 AWS Budgets 動作來建立預算。您也可以為選取的 AWS 資源建立已定義的預算臨界值警示。如需詳細資訊，請參閱 [《AWS Budgets 使用指南》中的「設定預算動作」](#)。同樣，您也可以使 CloudWatch 用 Amazon 創建帳單警報。如需詳細資訊，請參閱 [建立帳單警示來監控預估的 AWS 費用](#)。

## 最佳做法：將叢集移至新的次要版本或 AWS ParallelCluster 修補程式版本

目前，每個 AWS ParallelCluster 次要版本與其 `pcluster` CLI 一起都是獨立的。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 CLI 重新建立叢集。

若要將叢集移至新的次要或修補程式版本的程序最佳化，建議您執行下列動作：

- 將個人資料儲存在叢集外部建立的外部磁碟區中，例如 Amazon EFS 和 FSx for 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。
- 使用下列類型建立共用儲存系統。您可以使用 AWS CLI 或建立這些系統 AWS Management Console。
  - [SharedStorage / EbsSettings / VolumeId](#)
  - [SharedStorage / EfsSettings / FileSystemId](#)
  - [SharedStorage / FsxLustreSettings / FileSystemId](#)

將叢集配置中的檔案系統或磁碟區定義為現有檔案系統或磁碟區。如此一來，當您刪除叢集時，它們就會保留下來，並且可以連接到新叢集。

我們建議您將 Amazon EFS 或 FSx 用於 Lustre 檔案系統。這兩個系統都可以同時連接到多個叢集。此外，您可以在刪除現有叢集之前，將這些系統中的任何一個附加到新叢集。

- 使用 [自訂啟動程序動作](#)來自訂執行個體，而不是使用自訂 AMI。相反，如果您使用自定義 AMI，則需要為每個新版本發行刪除並重新創建該 AMI。
- 我們建議您依照下列順序套用前述建議：
  1. 更新現有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。測試新叢集時，請檢查下列項目：

- 請確定您的資料可在新叢集中使用。
  - 請確定您的應用程式可在新叢集中運作。
4. 在您的新叢集經過完整測試並運作且不再需要現有叢集之後，請將其刪除。

## 從 AWS ParallelCluster 2.x 移動到 3.x

### 自訂引導操作

使用 AWS ParallelCluster 3 時，您可以使用和 [Scheduling/SlurmQueues](#) 區段中的參數 (在版本 2 中) 和 `OnNodeStart` (`pre_install` 在 AWS ParallelCluster 版本 2 中)，為頭節點和 `OnNodeConfigured` 計算節點指定不同的自訂啟動程序動作指令碼。 `post_install` AWS ParallelCluster [HeadNode](#) 如需詳細資訊，請參閱 [自訂引導動作](#)。

為 AWS ParallelCluster 2 開發的自定義引導操作腳本必須適應在 AWS ParallelCluster 3 中使用：

- 我們不建議使用 `/etc/parallelcluster/cfnconfig` 和 `cfn_node_type` 區分主節點和計算節點。相反地，我們建議您在 [HeadNode](#) 和 [Scheduling/](#) 中指定兩個不同的指令碼 [SlurmQueues](#)。
- 如果您想要繼續載入 `/etc/parallelcluster/cfnconfig` 以在啟動程序動作指令碼中使用，請注意的值 `cfn_node_type` 已從 "MasterServer" 變更為 "HeadNode" (請參閱: [包容性語言](#))。
- 在 AWS ParallelCluster 2 上，引導動作指令碼的第一個輸入引數是指令碼的 S3 URL 並保留。在 AWS ParallelCluster 3 中，只有在配置中配置的參數傳遞給腳本。

#### Warning

不支持使用通過 `/etc/parallelcluster/cfnconfig` 文件提供的內部變量。此檔案可能會在 future 版本中移除。

## AWS ParallelCluster 2.x 和 3.x 使用不同的配置文件語法

AWS ParallelCluster 3.x 配置使用 YAML 語法。您可以在以下位置找到完整參考資料 [組態檔案](#)。

除了需要 YAML 檔案格式之外，已在 AWS ParallelCluster 3.x 中更新了許多組態區段、設定和參數值。在本節中，我們會注意到 AWS ParallelCluster 組態的主要變更，以及說明每個版本之間的差異的 side-by-side AWS ParallelCluster 範例。

### 啟用和停用超執行緒的多重排程器佇列組態範例

## AWS ParallelCluster 2:

```
[cluster default]
queue_settings = ht-enabled, ht-disabled
...

[queue ht-enabled]
compute_resource_settings = ht-enabled-i1
disable_hyperthreading = false

[queue ht-disabled]
compute_resource_settings = ht-disabled-i1
disable_hyperthreading = true

[compute_resource ht-enabled-i1]
instance_type = c5n.18xlarge
[compute_resource ht-disabled-i1]
instance_type = c5.xlarge
```

## AWS ParallelCluster 3:

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: ht-enabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-enabled-i1
          DisableSimultaneousMultithreading: true
          InstanceType: c5n.18xlarge
    - Name: ht-disabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-disabled-i1
          DisableSimultaneousMultithreading: false
          InstanceType: c5.xlarge
```

## Lustre 檔案系統組態的新 FSx 範例

## AWS ParallelCluster 2:

```
[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
storage_capacity = 1200
imported_file_chunk_size = 1024
import_path = s3://bucket
export_path = s3://bucket/export_dir
weekly_maintenance_start_time = 3:02:30
deployment_type = PERSISTENT_1
data_compression_type = LZ4
```

## AWS ParallelCluster 3:

```
...
SharedStorage:
- Name: fsx
  MountDir: /shared-fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    ImportedFileChunkSize: 1024
    ImportPath: s3://bucket
    ExportPath: s3://bucket/export_dir
    WeeklyMaintenanceStartTime: "3:02:30"
    DeploymentType: PERSISTENT_1
    DataCompressionType: LZ4
```

為 Lustre 檔案系統掛載現有 FSx 的叢集配置範例

## AWS ParallelCluster 2:

```
[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
```



```
fsx_fs_id = fsx_fs_id
```

### AWS ParallelCluster 3:

```
...
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      FileSystemId: fsx_fs_id
```

### 含有 Intel HPC 作業平台規格軟體堆疊的叢集範例

### AWS ParallelCluster 2:

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

### AWS ParallelCluster 3:

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

### 備註：

- Intel HPC 平台規格軟體的安裝必須遵守適用的 [Intel 使用者授權合約](#) 的條款與條件。

自訂 IAM 組態範例，包括：執行個體設定檔、執行個體角色、執行個體的其他政策，以及與叢集相關聯之 lambda 函數的角色

### AWS ParallelCluster 2:

```
[cluster default]
additional_iam_policies = arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess,arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
ec2_iam_role = ec2_iam_role
iam_lambda_role = lambda_iam_role
```

```
...
```

### AWS ParallelCluster 3:

```
...
Iam:
  Roles:
    CustomLambdaResources: lambda_iam_role
HeadNode:
  ...
  Iam:
    InstanceRole: ec2_iam_role
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ...
      Iam:
        InstanceProfile: iam_instance_profile
    - Name: queue2
      ...
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
          - Policy: arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
```

### 備註：

- 對於 AWS ParallelCluster 2，IAM 設定會套用至叢集的所有執行個體，且 `additional_iam_policies` 無法搭配使用 `ec2_iam_role`。
- 對於 AWS ParallelCluster 3，您可以為頭節點和運算節點設定不同的 IAM 設定，甚至可以為每個運算佇列指定不同的 IAM 設定。
- 對於 AWS ParallelCluster 3，您可以使用 IAM 執行個體設定檔做為 IAM 角色的替代方案。InstanceProfile，InstanceRole 或不 AdditionalIamPolicies 能一起配置。

### 自訂啟動程序動作範例

#### AWS ParallelCluster 2:

```
[cluster default]
s3_read_resource = arn:aws:s3:::bucket_name/*
```

```
pre_install = s3://bucket_name/scripts/pre_install.sh
pre_install_args = 'R curl wget'
post_install = s3://bucket_name/scripts/post_install.sh
post_install_args = "R curl wget"
...
```

### AWS ParallelCluster 3:

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

具有 S3 儲存貯體資源讀取和寫入存取權的叢集範例

### AWS ParallelCluster 2:

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

### AWS ParallelCluster 3:

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

## 包容性語言

AWS ParallelCluster 3 在 AWS ParallelCluster 2 中使用「主節點」的地方使用「頭節點」一詞。這包含下列項目：

- 在 AWS Batch 工作環境中匯出的變數已變更:從變更MASTER\_IP為PCLUSTER\_HEAD\_NODE\_IP。
- 所有 AWS CloudFormation 輸出都從變更Master\*為HeadNode\*。
- 全部 NodeType 和標籤從變更Master為HeadNode。

## 排程器 Support

AWS ParallelCluster 3.x 不支持網格引擎 ( SGE ) 和扭矩調度器的兒子。

awsbsub這些指 AWS Batch 令awsbhostsawsbkill、awsbout、awsbqueuesawsbstat、和作為單獨的 aws-parallelcluster-awsbatch-cli PyPI 套件散佈。此套件是由 AWS ParallelCluster 在頭節點上安裝的。您仍然可以從叢集的頭節點使用這些 AWS Batch 指令。但是，如果您希望從頭節點以外的位置使用 AWS Batch 命令，則必須先安裝 aws-parallelcluster-awsbatch-cli PyPI 套件。

## AWS ParallelCluster CLI

命 AWS ParallelCluster 令列介面 (CLI) 已變更。有關新語法的說明，請參閱[AWS ParallelCluster CLI 命令](#)。CLI 的輸出格式為 [JSON](#) 字串。

### 配置新叢集

與 AWS ParallelCluster 2 相比，該pcluster configure命令在 AWS ParallelCluster 3 中包含不同的參數。如需詳細資訊，請參閱[pcluster configure](#)。

另請注意，配置文件語法已從 AWS ParallelCluster 2 更改。如需叢集配置設定的完整參考資料，請參閱[叢集配置檔](#)。

### 建立新叢集

AWS ParallelCluster 2 的pcluster create命令已被[pcluster create-cluster](#)命令替換。

請注意 AWS ParallelCluster 2.x 中的默認行為 ( 沒有選-nw項 ) 是等待集群創建事件，而 AWS ParallelCluster 3.x 命令立即返回。您可以使用來監視叢集建立進度[pcluster describe-cluster](#)。

AWS ParallelCluster 3 配置文件包含單個集群定義，因此不再需要該-t參數。

以下是範例組態檔案。

```
# AWS ParallelCluster v2
$ pcluster create \
  -r REGION \
  -c V2_CONFIG_FILE \
  -nw \
```

```
-t CLUSTER_TEMPLATE \  
CLUSTER_NAME  
  
# AWS ParallelCluster v3  
$ pcluster create-cluster \  
  --region REGION \  
  --cluster-configuration V3_CONFIG_FILE \  
  --cluster-name CLUSTER_NAME
```

## 列出叢集

`pcluster list` AWS ParallelCluster 2.x 指令必須以指 [pcluster list-clusters](#) 令取代。

備註：您需要 AWS ParallelCluster v2 CLI 列出以 .x 版本建立的 AWS ParallelCluster 叢集。如 [AWS ParallelCluster 在虛擬環境中安裝 \(建議使用\)](#) 需如何安裝使 AWS ParallelCluster 用虛擬環境的多個版本，請參閱。

```
# AWS ParallelCluster v2  
$ pcluster list -r REGION  
  
# AWS ParallelCluster v3  
$ pcluster list-clusters --region REGION
```

## 啟動和停止叢集

`pcluster start` 和 `pcluster stop` AWS ParallelCluster 2.x 指令必須以指 [pcluster update-compute-fleet](#) 令取代。

啟動運算叢集：

```
# AWS ParallelCluster v2  
$ pcluster start \  
  -r REGION \  
  CLUSTER_NAME  
  
# AWS ParallelCluster v3 - Slurm fleets  
$ pcluster update-compute-fleet \  
  --region REGION \  
  --cluster-name CLUSTER_NAME \  
  --status START_REQUESTED
```

```
# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status ENABLED
```

停止運算叢集：

```
# AWS ParallelCluster v2
$ pcluster stop \
  -r REGION \
  CLUSTER_NAME

# AWS ParallelCluster v3 - Slurm fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status STOP_REQUESTED

# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status DISABLED
```

連線至叢集

`pcluster ssh` AWS ParallelCluster 2.x 指令在 AWS ParallelCluster 3.x 中有不同的參數名稱。請參閱 [pcluster ssh](#)。

連線至叢集：

```
# AWS ParallelCluster v2
$ pcluster ssh \
  -r REGION \
  CLUSTER_NAME \
  -i ~/.ssh/id_rsa

# AWS ParallelCluster v3
$ pcluster ssh \
  --region REGION \
  --cluster-name CLUSTER_NAME \
```

```
-i ~/.ssh/id_rsa
```

## IMDS 組態設定更新

從 3.0.0 版開始，依預 AWS ParallelCluster 設，支援將頭節點的 IMDS (以及執行個體設定檔認證) 存取權限制為超級使用者子集。如需更多詳細資訊，請參閱 [Imds 屬性](#)。

## AWS ParallelCluster 支援的區域

AWS ParallelCluster 版本 3 可用於以下內容 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
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



區域名稱	區域
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
以色列 (特拉維夫)	il-central-1

# 使用 AWS ParallelCluster

## 主題

- [AWS ParallelCluster 界面](#)
- [AWS Lambda 中的 VPC 組態 AWS ParallelCluster](#)
- [AWS Identity and Access Management 權限 AWS ParallelCluster](#)
- [網路組態](#)
- [登入節點](#)
- [自訂引導動作](#)
- [使用 Amazon S3](#)
- [使用 競價型執行個體](#)
- [排程器支援 AWS ParallelCluster](#)
- [共用儲存](#)
- [AWS ParallelCluster 資源和標記](#)
- [監控 AWS ParallelCluster 和記錄](#)
- [AWS CloudFormation 自訂資源](#)
- [Elastic Fabric Adapter](#)
- [啟用英特爾 MPI](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster 對於地形](#)
- [通過 NICE DCV Connect 到頭節點](#)
- [使用 pcluster update-cluster](#)
- [AWS ParallelCluster AMI 定制](#)
- [使用隨需容量保留 \(ODCR\) 啟動執行個體](#)
- [使用容量區塊 \(CB\) 啟動執行個體](#)
- [AMI 修補和 Amazon EC2 實例替換](#)
- [作業系統](#)

# AWS ParallelCluster 界面

AWS ParallelClusterUI 是一個基於 Web 的用戶界面，可作為用於創建，監視和管理集群的儀表板。您可以在AWS ParallelCluster您的AWS 帳戶。該AWS ParallelCluster用戶界面添加了AWS ParallelCluster版本 3.5.0。

若要安裝AWS ParallelCluster UI 並開始使用，請參閱[安裝AWS ParallelCluster使用者介面](#)和[使 AWS ParallelCluster 用 UI 配置和創建集群](#)。

The screenshot displays the AWS ParallelCluster user interface. At the top, the header shows 'AWS ParallelCluster' with a user profile 'user@domain.com' and the region 'eu-west-1'. The main navigation menu on the left includes 'Clusters', 'Images', and 'Users'. The central area is titled 'Clusters (2)' and contains a search bar and a table of clusters. The table lists two clusters: 'hpc-cluster-1' with a status of 'CREATE COMPLETE' and version '3.5.0', and 'hpc-cluster-2' with a status of 'DELETE IN PROGRESS' and version '3.5.0'. Below the table, the 'Cluster: hpc-cluster-1' details are shown, including tabs for 'Details', 'Instances', 'Storage', 'Job scheduling', and 'Stack events'. The 'Properties' section is expanded, showing configuration details, SSH command, EC2 Instance Connect, cluster status, compute fleet status, version, region, created time, and latest update time.

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

**Cluster: hpc-cluster-1**

**Properties**

- Cluster configuration**  
VIEW
- SSH command** Info  
ssh ec2-user@54.78.245.22
- EC2 Instance Connect** Info  
mssh -r eu-west-1 ec2-user@i-0b14dc1a2f5dc048e
- Cluster status**  
CREATE COMPLETE
- Compute fleet status**  
RUNNING
- Version**  
3.5.0
- Region**  
eu-west-1
- Created time**  
March 10, 2023 at 09:39 (UTC+1:00)
- Latest update time**  
March 10, 2023 at 09:39 (UTC+1:00)

AWS ParallelCluster使用者介面支援下列功能：

- 會顯示下列項目：
  - 您在使用中建立的叢集清單AWS 帳戶單AWS ParallelCluster。
  - 所列叢集的可用狀態和詳細資料。
  - CloudFormation 可用於監視的堆疊事件和AWS ParallelCluster記錄檔。

- 叢集上執行的工作狀態。
- 您可以將其用於建置叢集的自訂映像。
- UI 用來建立叢集的官方映像檔清單。
- 可以存取AWS ParallelCluster UI 的使用者清單。您可以新增和移除使用者。
- 提供 step-by-step 供建立和編輯 (更新) 叢集，以及選取要新增、編輯或移除的支援叢集功能的指引。對於正在編輯的叢集配置，無法變更無法存取的輸入欄位。您可以選擇在叢集部署之前對叢集配置執行乾式執行驗證。
- 具有直接殼層連結，可存取叢集檢視中的頭節點。在 step-by-step 指引期間選擇 [新增 SSM 工作階段]，以新增直接殼層存取權，以及在頭節點上新增 SSM 代管執行個體核心原則。

使用AWS ParallelCluster UI 建置和管理叢集時，請考量下列事項。

- 您只能建立和編輯叢集，或使用建立AWS ParallelCluster UI 的相同AWS ParallelCluster版本建立映像檔。早期版本的叢集或映像只能檢視。如果您管理多個版本的叢集和映像檔，建議您建立AWS ParallelCluster UI 執行個體以支援每個版本。
- AWS ParallelCluster使用者介面是專為鏡像pccluster CLI 功能而設計。有一些差異。如果您與 step-by-step 指引保持一致，則表示您正在使用所有支援的功能。在部署之前，您可以選擇手動編輯叢集或映像配置。如果您這樣做，我們建議您選擇「乾式運行」來驗證配置，以驗證您的編輯是否完全受到支援。

#### Note

AWS ParallelCluster使用者介面不支援AWS Batch。

## AWS Lambda 中的 VPC 組態 AWS ParallelCluster

AWS ParallelCluster 用 AWS Lambda 於在叢集的生命週期內執行作業。[AWS Lambda 函數一律會在 Lambda 服務擁有的 VPC](#) 中執行。此 Lambda 函數也可以連接到虛擬私有雲 (VPC) 中的私有子網路，以存取私有資源。

#### Note

Lambda 函數無法透過專用執行個體租用直接連線到 VPC。若要連線到專用 VPC 中的資源，請將專用 VPC 對等至具有可連線到專用 VPC 的預設租用的第二個 VPC。

如需詳細資訊，請參閱 Amazon EC2 Linux [執行個體使用者指南中的專用執行個體](#)和[如何將 Lambda 函數連接到專用 VPC](#)？從AWS 知識中心。


由建立的 Lambda 函數 AWS ParallelCluster 可以連接到私有 VPC。這些 Lambda 函數需要存取 AWS 服務。您可以使用下列方法，透過網際網路或 VPC 端點提供存取權。

- 互聯網接入

若要存取網際網路 AWS 服務，Lambda 函數需要網路位址轉譯 (NAT)。將輸出流量從您的私有子網路路由到公有子網路中的 [NAT 閘道](#)。

- VPC 端點

有數種 AWS 服務提供 [VPC 端點](#)。您可以使用 VPC 端點 AWS 服務 從無法存取網際網路的 VPC 連線到。若要檢視 AWS ParallelCluster VPC 端點清單，請參閱[網路](#)。

 Note

子網路和安全性群組的每個組合都必須提供 AWS 服務 使用這些方法之一的存取權。子網路和安全性群組必須位於相同的 VPC 中。

如需詳細資訊，請參閱 Amazon [Virtual Private Cloud 使用者指南中的 VPC 端點](#)和開發人員指南中針對虛擬私人雲端連線功能的[網際網路和服務存取](#)。AWS Lambda

若要設定 Lambda 函數和 VPC 的使用方式，請參閱叢集[LambdaFunctionsVpcConfig](#)的 [DeploymentSettings](#)/或映像檔[LambdaFunctionsVpcConfig](#)的 [DeploymentSettings](#)/。

## AWS Identity and Access Management 權限 AWS ParallelCluster

AWS ParallelCluster 在建立和管理叢集時，使用 IAM 許可來控制對資源的存取。

若要在 AWS 帳戶中建立和管理叢集，AWS ParallelCluster 需要兩個層級的權限：

- `pcluster`使用者呼叫 `pcluster` CLI 命令以建立和管理叢集所需的權限。
- 叢集資源執行叢集動作所需的權限。

AWS ParallelCluster 使用 [Amazon EC2 執行個體設定檔和角色](#) 提供叢集資源許可。若要管理叢集資源許可 AWS ParallelCluster 可，還需要 IAM 資源的許可。如需詳細資訊，請參閱 [AWS ParallelCluster 管理 IAM 資源的使用者範例政策](#)。

pcluster 使用者需要 IAM 許可才能使用 [pcluster](#) CLI 建立和管理叢集及其資源。這些許可包含在 IAM 政策中，可新增至使用者或角色。如需 IAM 角色的詳細資訊，請參閱 [使用者指南中的建立使 AWS Identity and Access Management 用者角色](#)。

您也可以使用 [AWS ParallelCluster 用於管理 IAM 許可的組態參數](#)。

以下各節包含必要的權限和範例。

若要使用範例原則，請以適當的值取代 `<REGION><AWS ACCOUNT ID>`、和類似字串。

下列範例政策包括資源的 Amazon 資源名稱 (ARN)。如果您在 AWS GovCloud (US) 或中 AWS 國分區中工作，則必須更改 ARN。具體來說，它們必須從分割區的「arn:aws」變更為「arn:aws-us-gov」，或將中國 AWS GovCloud (US) 分割區的「arn:aws-cn」變更為「arn:」。AWS 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 資源名稱 \(ARN\) 和中國 AWS 服務入門中的中國 AWS 服務的 ARN](#)。

您可以在 [上的 AWS ParallelCluster 文件](#) 中追蹤範例政策的變更 GitHub。

## 主題

- [AWS ParallelCluster Amazon EC2 實例角色](#)
- [AWS ParallelCluster pcluster 使用者策略範例](#)
- [AWS ParallelCluster 管理 IAM 資源的使用者範例政策](#)
- [AWS ParallelCluster 用於管理 IAM 許可的組態參數](#)

## AWS ParallelCluster Amazon EC2 實例角色

使用預設組態設定建立叢集時，會 AWS ParallelCluster 使用 Amazon [EC2 執行個體設定檔](#) 自動建立預設叢集 Amazon [EC2 執行個體角色](#)，以提供建立和管理叢集及其資源所需的許可。

### 使用預設 AWS ParallelCluster 執行個體角色的替代方法

您可以使用 InstanceRole 叢集組態設定來代替預設 AWS ParallelCluster 執行個體角色，為 EC2 指定自己的現有 IAM 角色。如需詳細資訊，請參閱 [AWS ParallelCluster 用於管理 IAM 許可的組態參數](#)。一般而言，您可以指定現有的 IAM 角色，以完全控制授與 EC2 的許可。

如果您想要將額外政策新增至預設執行個體角色，建議您使用[AdditionalIamPolicies](#)組態設定而非[InstanceProfile](#)或[InstanceRole](#)設定來傳遞額外的 IAM 政策。您可以在更新叢集AdditionalIamPolicies時進行更新，但是在更新叢集InstanceRole時無法更新。

## AWS ParallelCluster **pcluster**使用者策略範例

下列範例顯示使用 pcluster CLI 建立 AWS ParallelCluster 和管理所需的使用者原則及其資源。您可以將策略附加到使用者或角色。

### 主題

- [基本 AWS ParallelCluster pcluster使用者原則](#)
- [使用 AWS Batch 排程器時的其他使用 AWS ParallelCluster pcluster者原](#)
- [使用亞馬遜 FSx 進行光澤時的其他用 AWS ParallelCluster pcluster戶政策](#)
- [AWS ParallelCluster 圖像構建pcluster用戶策略](#)

## 基本 AWS ParallelCluster **pcluster**使用者原則

下列原則顯示執行 AWS ParallelCluster pcluster命令所需的權限。

包括原則中列出的最後一個動作，以驗證叢集配置中指定的任何密碼。例如，AWS Secrets Manager 密碼用於配置[DirectoryService](#)整合。在此情況下，只有在中存在有效密碼時，才會建立叢集[PasswordSecretArn](#)。如果省略此動作，則會略過密碼驗證。若要改善您的安全性狀態，建議您僅新增叢集配置中指定的密碼，以縮小此原則陳述式的範圍。

### Note

如果現有的 Amazon EFS 檔案系統是叢集中唯一使用的檔案系統，您可以將 Amazon EFS 政策陳述式範例範圍縮減為叢集組態檔中參照[SharedStorage](#) 區段的特定檔案系統。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow",
```

```

    "Sid": "EC2Read"
  },
  {
    "Action": [
      "ec2:AllocateAddress",
      "ec2:AssociateAddress",
      "ec2:AttachNetworkInterface",
      "ec2:AuthorizeSecurityGroupEgress",
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:CreateFleet",
      "ec2:CreateLaunchTemplate",
      "ec2:CreateLaunchTemplateVersion",
      "ec2:CreateNetworkInterface",
      "ec2:CreatePlacementGroup",
      "ec2:CreateSecurityGroup",
      "ec2:CreateSnapshot",
      "ec2:CreateTags",
      "ec2>DeleteTags",
      "ec2:CreateVolume",
      "ec2>DeleteLaunchTemplate",
      "ec2>DeleteNetworkInterface",
      "ec2>DeletePlacementGroup",
      "ec2>DeleteSecurityGroup",
      "ec2>DeleteVolume",
      "ec2:DisassociateAddress",
      "ec2:ModifyLaunchTemplate",
      "ec2:ModifyNetworkInterfaceAttribute",
      "ec2:ModifyVolume",
      "ec2:ModifyVolumeAttribute",
      "ec2:ReleaseAddress",
      "ec2:RevokeSecurityGroupEgress",
      "ec2:RevokeSecurityGroupIngress",
      "ec2:RunInstances",
      "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Write"
  },
  {
    "Action": [
      "dynamodb:DescribeTable",
      "dynamodb:ListTagsOfResource",
      "dynamodb:CreateTable",

```



```

        "dynamodb:DeleteTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:Query",
        "dynamodb:TagResource"
    ],
    "Resource": "arn:aws:dynamodb:*:<AWS ACCOUNT ID>:table/parallelcluster-*",
    "Effect": "Allow",
    "Sid": "DynamoDB"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:CreateHostedZone",
        "route53>DeleteHostedZone",
        "route53:GetChange",
        "route53:GetHostedZone",
        "route53:ListResourceRecordSets",
        "route53:ListQueryLoggingConfigs"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
},
{
    "Action": [
        "cloudformation:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudFormation"
},
{
    "Action": [
        "cloudwatch:PutDashboard",
        "cloudwatch:ListDashboards",
        "cloudwatch>DeleteDashboards",
        "cloudwatch:GetDashboard",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch>DeleteAlarms",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutCompositeAlarm"
    ]
}

```

```

    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatch"
  },
  {
    "Action": [
      "iam:GetRole",
      "iam:GetRolePolicy",
      "iam:GetPolicy",
      "iam:SimulatePrincipalPolicy",
      "iam:GetInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:policy/*",
      "arn:aws:iam::aws:policy/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRead"
  },
  {
    "Action": [
      "iam:CreateInstanceProfile",
      "iam>DeleteInstanceProfile",
      "iam:AddRoleToInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamInstanceProfile"
  },
  {
    "Condition": {
      "StringEqualsIfExists": {
        "iam:PassedToService": [
          "lambda.amazonaws.com",
          "ec2.amazonaws.com",
          "spotfleet.amazonaws.com"
        ]
      }
    }
  }
}

```

```

    },
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda:DeleteFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:GetFunction",
        "lambda:InvokeFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:UpdateFunctionConfiguration",
        "lambda:TagResource",
        "lambda:ListTags",
        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:parallelcluster-*",
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "s3:*"
    ],
    "Resource": [
        "arn:aws:s3:::parallelcluster-*",
        "arn:aws:s3:::aws-parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
},
{
    "Action": [

```

```

        "s3:Get*",
        "s3:List*"
    ],
    "Resource": "arn:aws:s3:::*-aws-parallelcluster*",
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
},
{
    "Action": [
        "elasticfilesystem:*"
    ],
    "Resource": [
        "arn:aws:elasticfilesystem:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "EFS"
},
{
    "Action": [
        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy",
        "logs:DescribeLogGroups",
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:CreateExportTask",
        "logs:DescribeLogStreams",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:PutMetricFilter",
        "logs>DeleteMetricFilter"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogs"
},
{
    "Action": [
        "resource-groups:ListGroupResources"
    ],
    "Resource": "*",
    "Effect": "Allow",

```

```

    "Sid": "ResourceGroupRead"
  },
  {
    "Sid": "AllowDescribingFileCache",
    "Effect": "Allow",
    "Action": [
      "fsx:DescribeFileCaches"
    ],
    "Resource": "*"
  },
  {
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET
NAME>",
    "Effect": "Allow"
  }
]
}

```

## 使用 AWS Batch 排程器時的其他使用 AWS ParallelCluster **pcluster** 者原

如果您需要使用 AWS Batch 排程器建立和管理叢集，則需要下列額外原則。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEqualsIfExists": {
          "iam:PassedToService": [
            "ecs-tasks.amazonaws.com",
            "batch.amazonaws.com",
            "codebuild.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
    }
  ]
}

```

```

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

```

```

    "Action": [
      "events:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "AmazonCloudWatchEvents"
  },
  {
    "Action": [
      "ecs:DescribeContainerInstances",
      "ecs:ListContainerInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ECS"
  }
]
}

```

## 使用亞馬遜 FSx 進行光澤時的其他用 AWS ParallelCluster `pcluster` 用戶政策

如果您需要使用適用於 Lustre 的 Amazon FSx 建立和管理叢集，則需要以下額外政策。

### Note

如果現有 Amazon FSx 檔案系統是叢集中唯一使用的檔案系統，您可以將 Amazon FSx 政策陳述式範例範圍縮減為叢集組態檔中參照的 [SharedStorage 區段](#) 特定檔案系統。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "fsx.amazonaws.com",
            "s3.data-source.lustre.fsx.amazonaws.com"
          ]
        }
      },
      "Action": [

```

```

        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "fsx:*"
    ],
    "Resource": [
        "arn:aws:fsx:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "FSx"
},
{
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/*",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:Get*",
        "s3:List*",
        "s3:PutObject"
    ],
    "Resource": "arn:aws:s3:::<S3 NAME>",
    "Effect": "Allow"
}
]
}

```

## AWS ParallelCluster 圖像構建 **pcluster** 用戶策略

打算使用建立自訂 Amazon EC2 映像的使用者 AWS ParallelCluster 必須具有以下一組許可。

```
{
```



```

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

```

```

        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IAMPassRole"
},
{
    "Action": [
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs>DeleteLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/imagebuilder/
ParallelClusterImage-*",
        "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/lambda/
ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "CloudWatch"
},
{
    "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation>CreateStack",
        "cloudformation>DeleteStack"
    ],
    "Resource": [
        "arn:aws:cloudformation:*:<AWS ACCOUNT ID>:stack/*"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormation"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda:GetFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda>DeleteFunction",
        "lambda:TagResource",
        "lambda>ListTags",
        "lambda:UntagResource"

```

```

    ],
    "Resource": [
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "imagebuilder:Get*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ImageBuilderGet"
},
{
    "Action": [
        "imagebuilder:CreateImage",
        "imagebuilder:TagResource",
        "imagebuilder:CreateImageRecipe",
        "imagebuilder:CreateComponent",
        "imagebuilder:CreateDistributionConfiguration",
        "imagebuilder:CreateInfrastructureConfiguration",
        "imagebuilder>DeleteImage",
        "imagebuilder>DeleteComponent",
        "imagebuilder>DeleteImageRecipe",
        "imagebuilder>DeleteInfrastructureConfiguration",
        "imagebuilder>DeleteDistributionConfiguration"
    ],
    "Resource": [
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image/parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:component/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:distribution-configuration/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:infrastructure-configuration/
parallelclusterimage-*"
    ],
    "Effect": "Allow",
    "Sid": "ImageBuilder"
},
{

```

```

    "Action": [
      "s3:CreateBucket",
      "s3:ListBucket",
      "s3:ListBucketVersions"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3Bucket"
  },
  {
    "Action": [
      "sns:GetTopicAttributes",
      "sns:TagResource",
      "sns:CreateTopic",
      "sns:Subscribe",
      "sns:Publish",
      "SNS:DeleteTopic",
      "SNS:Unsubscribe"
    ],
    "Resource": [
      "arn:aws:sns:*:<AWS ACCOUNT ID>:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "SNS"
  },
  {
    "Action": [
      "s3:PutObject",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "S3Objects"
  },
  {
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",

```

```
        "Resource": "arn:aws:iam::*:role/aws-service-role/  
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",  
        "Condition": {  
            "StringLike": {  
                "iam:AWSServiceName": "imagebuilder.amazonaws.com"  
            }  
        }  
    }  
]  
}
```

## AWS ParallelCluster 管理 IAM 資源的使用者範例政策

使用 AWS ParallelCluster 建立叢集或自訂 AMI 時，必須提供包含許可的 IAM 政策，才能將必要的許可集授與 AWS ParallelCluster 元件。這些 IAM 資源可由 AWS ParallelCluster 自動建立，也可以在建立叢集或自訂映像時作為輸入提供。

您可以使用下列模式，透過在組態中使用其他 IAM 政策，為使用 AWS ParallelCluster 者提供存取 IAM 資源所需的許可。

### 主題

- [特權 IAM 存取模式](#)
- [受限制的 IAM 存取模式](#)
- [PermissionsBoundary 模式](#)

### 特權 IAM 存取模式

使用此模式，AWS ParallelCluster 自動建立所有必要的 IAM 資源。這些 IAM 政策的範圍縮小為僅啟用叢集資源的存取權。

若要啟用「特權 IAM」存取模式，請將下列政策新增至使用者角色。

#### Note

如果您設定 [HeadNode/Iam/AdditionalPolicies](#) 或 [Scheduling//SlurmQueuesIam/AdditionalPolicies](#) 參數，則必須提供 AWS ParallelCluster 使用者附加和卸離每個其他原則的角色原則的權限，如下列原則所示。將其他原則 ARN 新增至附加和卸離角色原則的條件。

**⚠ Warning**

此模式可讓使用者在 AWS 帳戶

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

```

        "iam:PolicyARN": [
            "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
            "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
            "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
            "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
            "arn:aws:iam::aws:policy/AWSBatchFullAccess",
            "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
            "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",
            "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
            "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
            "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
            "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
            "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
        ]
    },
    "Action": [
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamPolicy"
}
]
}

```

## 受限制的 IAM 存取模式

如果沒有向使用者授與其他 IAM 政策，則叢集或自訂映像組建所需的 IAM 角色必須由管理員手動建立，並在叢集設定中傳遞。

建立叢集時，需要下列參數：

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

建置自訂映像檔時，需要下列參數：

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

作為上述參數一部分傳遞的 IAM 角色必須在 `/parallelcluster/` 路徑前綴上創建。如果無法執行此操作，則需要更新使用者原則以授與特定自訂角色的 `iam:PassRole` 權限，如下列範例所示。

```
{
  "Condition": {
    "StringEqualsIfExists": {
      "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "lambda.amazonaws.com",
        "ec2.amazonaws.com",
        "spotfleet.amazonaws.com",
        "batch.amazonaws.com",
        "codebuild.amazonaws.com"
      ]
    }
  },
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    <list all custom IAM roles>
  ],
  "Effect": "Allow",
  "Sid": "IamPassRole"
}
```

#### Warning

目前此模式不允許管理 AWS Batch 叢集，因為並非所有 IAM 角色都可以在叢集配置中傳遞。

## PermissionsBoundary 模式

此模式會委派建 AWS ParallelCluster 立 IAM 角色，這些角色繫結至已設定的 IAM 許可界限。如需 IAM 許可界限的詳細資訊，請參閱《IAM 使用者指南》中的 IAM [實體的許可界限](#)。



需要將下列原則新增至使用者角色。

在政策中，將 `< permissions-boundary-arn >` 取代為要強制執行為許可界限的 IAM 政策 ARN。

**⚠ Warning**

如果您設定 [HeadNode//AdditionalPolicies](#) 或 [Iam//SchedulingSlurmQueuesIam/AdditionalPolicies](#) 參數，則必須授與使用者為每個其他原則附加和中斷連結角色原則的權限，如下列原則所示。將其他原則 ARN 新增至附加和卸離角色原則的條件。

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

```

    "Effect": "Allow",
    "Sid": "IamCreateRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          <permissions-boundary-arn>
        ]
      }
    },
    "Action": [
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          <permissions-boundary-arn>
        ]
      }
    },
    "ArnLike": {
      "iam:PolicyARN": [
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
        "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
        "arn:aws:iam::aws:policy/AWSBatchFullAccess",
        "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
        "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
        "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
        "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
      ]
    }
  }
}

```

```
    ]
  },
  "Action": [
    "iam:AttachRolePolicy",
    "iam:DetachRolePolicy"
  ],
  "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
  "Effect": "Allow",
  "Sid": "IamPolicy"
}
]
```

啟用此模式時，您必須在建立或更新叢集時，在 [Iam/PermissionsBoundary](#) 配置參數中指定權限邊界 ARN，以及在建立自訂映像時在 [Build/Iam/PermissionBoundary](#) 參數中指定權限邊界 ARN。

## AWS ParallelCluster 用於管理 IAM 許可的組態參數

AWS ParallelCluster 公開一系列組態選項，以自訂和管理在叢集或自訂 AMI 建立程序期間使用的 IAM 許可和角色。

### 主題

- [叢集組態](#)
- [自定義映像配置](#)

## 叢集組態

### 主題

- [主節點 IAM 角色](#)
- [Amazon S3 訪問](#)
- [其他 IAM 政策](#)
- [AWS Lambda 函數角色](#)
- [運算節點 IAM 角色](#)
- [許可界限](#)

## 主節點 IAM 角色

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

使用此選項，您可以覆寫指派給叢集頭節點的預設 IAM 角色。有關其他詳細信息，請參閱[InstanceProfile](#)參考。

以下是當排程器為 Slurm 時，作為此角色一部分使用的最小原則集：

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` 受管理的 IAM 政策。如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的建立 IAM 角色和使用者[以搭配 CloudWatch 代理程式](#)使用。
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 受管理的 IAM 政策。如需詳細資訊，請參閱《AWS Systems Manager 使用指南》AWS Systems Manager 中的[AWS 受管理策略](#)。
- 其他 IAM 政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.<REGION>/*",
        "arn:aws:s3:::parallelcluster-*<v1-do-not-delete>*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:BatchGetItem"
      ],
      "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
    }
  ]
}
```

```
    "Effect": "Allow"
  },
  {
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/parallelcluster:node-type": "Compute"
      }
    },
    "Action": "ec2:TerminateInstances",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:RunInstances",
      "ec2:CreateFleet"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "ec2.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeVolumes",
      "ec2:DescribeInstanceAttribute",
      "ec2:DescribeCapacityReservations"
    ]
  }
}
```

```

    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:CreateTags",
      "ec2:AttachVolume"
    ],
    "Resource": [
      "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
      "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackResource",
      "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "route53:ChangeResourceRecordSets"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET_ID>",
    "Effect": "Allow"
  }
]
}

```

請注意，如果使用 [SchedulingSlurmQueuesIam](#) 或 [InstanceRole](#) 覆寫運算 IAM 角色，則上述報告的頭節點策略需要在 iam:PassRole 權限 Resource 部分中包含此類角色。

以下是當排程器為時，作為此角色一部分使用的最小原則集 AWS Batch：

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` 受管理的 IAM 政策。如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的建立 IAM 角色和使用者 [以搭配 CloudWatch 代理程式](#) 使用。
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 受管理的 IAM 政策。如需詳細資訊，請參閱《AWS Systems Manager 使用指南》AWS Systems Manager 中的 [AWS 受管理策略](#)。
- 其他 IAM 政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::dvc-license.<REGION>/*",
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "batch:DescribeJobQueues",
        "batch:DescribeJobs",
        "batch:ListJobs",
        "batch:DescribeComputeEnvironments"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "batch:SubmitJob",
        "batch:TerminateJob",
        "logs:GetLogEvents",
        "ecs:ListContainerInstances",
        "ecs:DescribeContainerInstances",
    ],
    "Resource": [
        "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-
stream:PclusterJobDefinition*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/AWSBatch-
PclusterComputeEnviron*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/AWSBatch-Pcluster*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/
PclusterJobQueue*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-definition/
PclusterJobDefinition*:*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeVolumes",

```



```

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

```

## Amazon S3 訪問

### [HeadNode/Iam/S3Access](#) 或 [SchedulingSlurmQueues/S3Access](#)

在這些組態區段中，您可以在建立此類角色時，將額外的 Amazon S3 政策授予與叢集的主節點或運算節點關聯的 IAM 角色，以自訂 Amazon S3 存取 AWS ParallelCluster。如需詳細資訊，請參閱每個組態參數的參考文件。

此參數只能在使用或設定使用者時使[特權 IAM 存取模式](#)用[PermissionsBoundary 模式](#)。

## 其他 IAM 政策

[HeadNode/Iam/AdditionalIamPolicies](#) 或 [SlurmQueuesIam/AdditionalIamPolicies](#)

使用此選項可在建立此類角色時，將其他受管 IAM 政策附加至與叢集的主節點或運算節點相關聯的 IAM 角色 AWS ParallelCluster。

### Warning

若要使用此選項，請確定已授 `iam:AttachRolePolicy` 與使用 [AWS ParallelCluster 者](#)，以及需要附加的 IAM 政策的 `iam:DetachRolePolicy` 許可。

## AWS Lambda 函數角色

[Iam / Roles / LambdaFunctionsRole](#)

此選項會覆寫附加至叢集建立程序期間所使用之所有 AWS Lambda 功能的角色。AWS Lambda 需要配置為允許擔任角色的主參與者。

### Note

如果設 [LambdaFunctionsVpcConfig](#) 定 [DeploymentSettings/](#)，則 `LambdaFunctionsRole` 必須包含 [AWS Lambda 角色權限](#) 才能設定 VPC 組態。

以下是作為此角色一部分使用的最小原則集：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "arn:aws:route53:::hostedzone/*",
      "Effect": "Allow"
    },
    {
```

```

    "Action": ["logs:CreateLogStream", "logs:PutLogEvents"],
    "Effect": "Allow",
    "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/lambda/
pcluster-*"
  },
  {
    "Action": "ec2:DescribeInstances",
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": "ec2:TerminateInstances",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/parallelcluster:node-type": "Compute"
      }
    },
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "s3:DeleteObject",
      "s3:DeleteObjectVersion",
      "s3:ListBucket",
      "s3:ListBucketVersions"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete",
      "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete/*"
    ]
  }
]
}

```

## 運算節點 IAM 角色

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

此選項允許覆寫指派給叢集運算節點的 IAM 角色。如需詳細資訊，請參閱 [InstanceProfile](#)。

以下是作為此角色一部分使用的最小原則集：

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` 受管理的 IAM 政策。如需詳細資訊，請參閱 Amazon 使用 CloudWatch 者指南中的建立 IAM 角色和使用者 [以搭配 CloudWatch 代理程式](#) 使用。
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 受管理的 IAM 政策。如需詳細資訊，請參閱《AWS Systems Manager 使用指南》AWS Systems Manager 中的 [AWS 受管理策略](#)。
- 其他 IAM 政策：

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

```
}
```

## 許可界限

### [Iam / PermissionsBoundary](#)

此參數會強制 AWS ParallelCluster 將指定的 IAM 政策作PermissionsBoundary為叢集部署一部分建立的所有 IAM 角色附加。

[PermissionsBoundary 模式](#)如需定義此設定時使用者所需的原則清單，請參閱。

## 自定義映像配置

### 主題

- [EC2 Image Builder 的執行個體角色](#)
- [AWS Lambda 清理角色](#)
- [其他 IAM 政策](#)
- [許可界限](#)

### EC2 Image Builder 的執行個體角色

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

使用此選項，您可以覆寫指派給 EC2 Image Builder 啟動之 Amazon EC2 執行個體的 IAM 角色，以建立自訂 AMI。

以下是作為此角色一部分使用的最小原則集：

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`受管理的 IAM 政策。如需詳細資訊，請參閱《AWS Systems Manager 使用指南》AWS Systems Manager中的[AWS 受管理策略](#)。
- `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder`受管理的 IAM 政策。如需詳細資訊，請參閱 Image Builder 使用指南中的[EC2InstanceProfileForImageBuilder政策](#)。
- 其他 IAM 政策：

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

```

```

    {
      "Action": [
        "ec2:CreateTags",
        "ec2:ModifyImageAttribute"
      ],
      "Resource": "arn:aws:ec2:<REGION>::image/*",
      "Effect": "Allow"
    }
  ]
}

```

## AWS Lambda 清理角色

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

此選項會覆寫附加至自訂映像檔建置程序期間所使用之所有 AWS Lambda 函數的角色。AWS Lambda 需要配置為允許擔任角色的主參與者。

#### Note

如果設 [LambdaFunctionsVpcConfig](#) 定 [DeploymentSettings/](#) , 則 [CleanupLambdaRole](#) 必須包含 [AWS Lambda 角色權限](#) 才能設定 VPC 組態。

以下是作為此角色一部分使用的最小原則集：

- `arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` 受管理的 IAM 政策。如需詳細資訊，請參閱 AWS Lambda 開發人員指南中的 [Lambda 功能AWS 受管政策](#)。
- 其他 IAM 政策：

```

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

```

```

    },
    {
      "Action": [
        "iam:DeleteInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
      "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/
parallelcluster/*",
      "Effect": "Allow"
    },
    {
      "Action": "imagebuilder:DeleteInfrastructureConfiguration",
      "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT
ID>:infrastructure-configuration/parallelclusterimage-*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "imagebuilder:DeleteComponent"
      ],
      "Resource": [
        "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:component/
parallelclusterimage-*/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "imagebuilder:DeleteImageRecipe",
      "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*/*",
      "Effect": "Allow"
    },
    {
      "Action": "imagebuilder:DeleteDistributionConfiguration",
      "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:distribution-
configuration/parallelclusterimage-*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "imagebuilder:DeleteImage",
        "imagebuilder:GetImage",
        "imagebuilder:CancelImageCreation"
      ],
    },
  ],

```

```

    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image/
parallelclusterimage-*/**",
    "Effect": "Allow"
  },
  {
    "Action": "cloudformation:DeleteStack",
    "Resource": "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*",
    "Effect": "Allow"
  },
  {
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:<REGION>::image/*",
    "Effect": "Allow"
  },
  {
    "Action": "tag:TagResources",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:DeleteFunction",
      "lambda:RemovePermission"
    ],
    "Resource": "arn:aws:lambda:<REGION>:<AWS ACCOUNT
ID>:function:ParallelClusterImage-*",
    "Effect": "Allow"
  },
  {
    "Action": "logs:DeleteLogGroup",
    "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/
lambda/ParallelClusterImage-*:*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "SNS:GetTopicAttributes",
      "SNS:DeleteTopic",
      "SNS:GetSubscriptionAttributes",
      "SNS:Unsubscribe"
    ],
    "Resource": "arn:aws:sns:<REGION>:<AWS ACCOUNT ID>:ParallelClusterImage-
**",
    "Effect": "Allow"
  }
}

```



```
    }  
  ]  
}
```

## 其他 IAM 政策

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

您可以使用此選項將其他受管 IAM 政策附加到與 EC2 Image Builder 用來產生自訂 AMI 之 Amazon EC2 執行個體相關聯的角色。

#### Warning

若要使用此選項，請確定已授iam:AttachRolePolicy與使用[AWS ParallelCluster](#)者，以及需要附加的 IAM 政策的iam:DetachRolePolicy許可。

## 許可界限

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

此參數強制 AWS ParallelCluster 將指定的 IAM 政策作為附加PermissionsBoundary到作為自訂 AMI 組建一部分建立的所有 IAM 角色。

如需使用此類功能所需的原則清單，請[PermissionsBoundary 模式](#)參閱。

## 網路組態

AWS ParallelCluster 使用 Amazon Virtual Private Cloud (VPC) 進行聯網。VPC 提供了一個靈活且可配置的網路平台，您可以在其中部署叢集。

VPC 必須具有 DNS Resolution = yes、DNS Hostnames = yes 及 DHCP 選項，並且具有區域的正確網域名稱。預設的 DHCP 選項組已指定所需的 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：

- 頭節點和運算節點都有一個子網路。
- 兩個子網路，頭節點位於一個公用子網路中，並計算私有子網路中的節點。子網路可以是新的子網路，也可以是現有子網路。

所有這些配置都可以在公共 IP 位址的情況下運行。AWS ParallelCluster 也可以部署為所有 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，您可以設定具有透過網際網路存取所有權限的單一公用子網路。或者，您可以使用 AWS Direct Connect 和 HTTP Proxy 為所有流量設定完全私人網路。

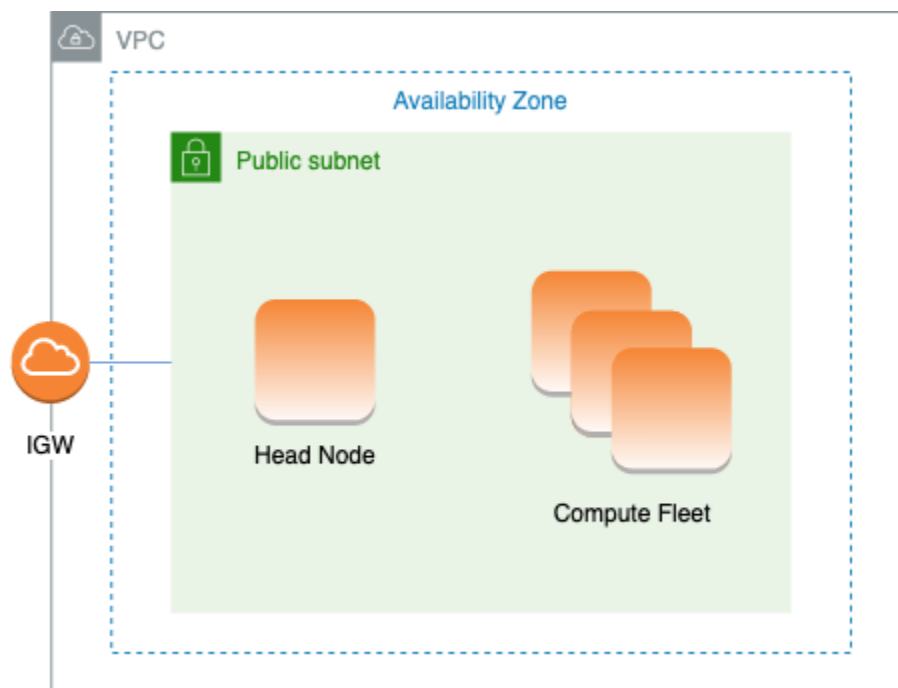
從 AWS ParallelCluster 3.0.0 開始 SecurityGroups, AdditionalSecurityGroups 可以為每個隊列配置不同的 PlacementGroup 設置。如需詳細資訊，請參閱 [HeadNodeSlurmQueues/Networking](#) 和 [Networking](#) 和 [AwsBatchQueues/Networking](#)。

如需某些網路案例的圖解，請參閱下列架構圖。

### 主題

- [AWS ParallelCluster 在單一公用子網路中](#)
- [AWS ParallelCluster 使用兩個子網](#)
- [AWS ParallelCluster 在使用連接的單個私有子網中 AWS Direct Connect](#)
- [AWS ParallelCluster 與 AWS Batch 調度程序](#)
- [AWS ParallelCluster 在沒有互聯網訪問的單個子網中](#)

## AWS ParallelCluster 在單一公用子網路中



此架構的組態需要下列設定：

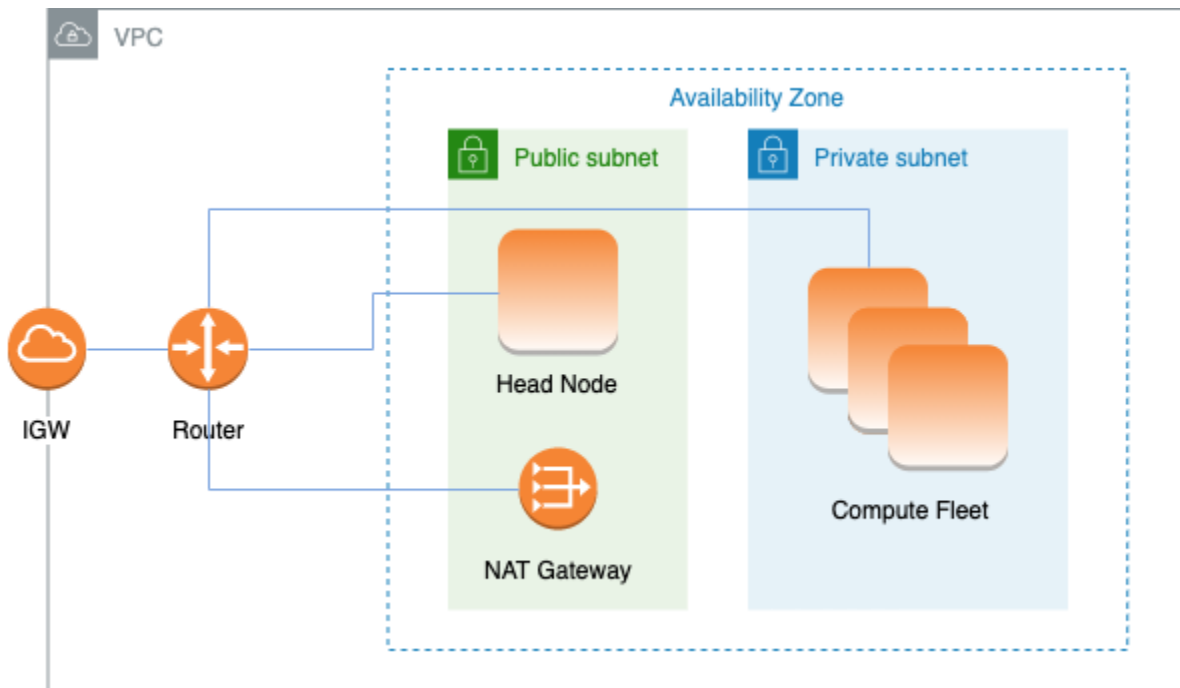
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
        - subnet-12345678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

在此配置中，叢集的所有執行個體都必須指派公用 IP，才能存取網際網路。若要完成此動作，請執行下列操作：

- 通過打開//中使用的子網路的「啟用自動分配公共 IPv4 地址」設置 [SubnetId](#) 或在 [HeadNode/Networking/](#) 中分配彈性 IP，以確保頭節點分配了公共 [HeadNodeIP](#) 地址。 [NetworkingElasticIp](#)
- 透過開啟///中使用的子網路的「啟用自動指派公用 IPv4 位址」設定， [SubnetIds](#) 或在 [Scheduling/SlurmQueuesNetworking/](#) 中設定 [AssignPublicIp](#) : true，以確保計算節點已指派公用 IP 位 [Scheduling](#) 址。 [SlurmQueuesNetworking](#)
- 如果您定義 p4d 執行個體類型，或其他具有多個網路介面或網路介面卡的執行個體類型至頭節點，則必須將 [HeadNode/Networking/](#) 設定 [ElasticIp](#) true 為提供公用存取權。AWS 只能將公有 IP 指派給透過單一網路介面啟動的執行個體。在此情況下，建議您使用 [NAT 閘道](#) 來提供叢集運算節點的公用存取權。[如需 IP 地址的詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公有 IPv4 地址。](#)
- 您無法定義 p4d 或 hp6id 執行個體類型，也無法定義具有多個網路介面或網路介面卡來計算節點的執行個體類型，因為只能將 AWS 公用 IP 指派給使用單一網路介面啟動的執行個體。[如需 IP 地址的詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公有 IPv4 地址。](#)

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



將現有私有子網路用於運算執行個體的組態需要下列設定：

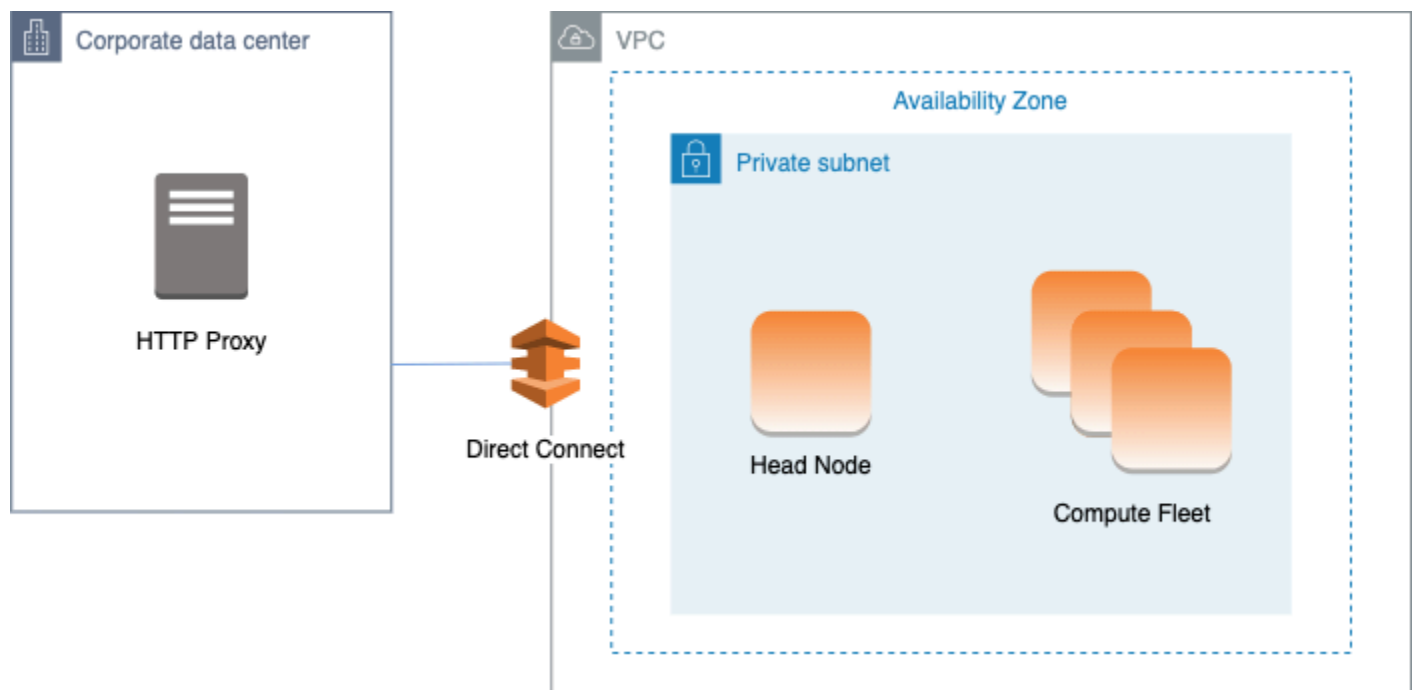
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
        - subnet-23456789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

在此配置中，只有叢集的頭節點才能指派公用 IP。您可以透過為//中使用的子網路開啟「啟用自動指派公用 IPv4 位址」設定，[SubnetId](#)或在 [HeadNode/Networking/](#)中指派彈性 IP 來達成此目的[HeadNode](#)。[NetworkingElasticIp](#)

如果您定義 p4d 執行個體類型或具有多個網路介面或網路介面卡的其他執行個體類型，則必須將 [HeadNode/Networking](#) 設定為提供 `ElasticIp` 用 `true` 取權。AWS 只能將公有 IP 指派給透過單一網路介面啟動的執行個體。[如需 IP 地址的詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公有 IPv4 地址。](#)

此設定需要 [NAT 閘道](#) 或佇列所使用之子網路中的內部 Proxy，才能提供運算執行個體的網際網路存取權。

## AWS ParallelCluster 在使用連接的單個私有子網中 AWS Direct Connect



此架構的組態需要下列設定：

```
# Note that all values are only provided as examples
HeadNode:
  ...
Networking:
  SubnetId: subnet-34567890 # subnet with proxy
  Proxy:
    HttpProxyAddress: http://proxy-address:port
Ssh:
  KeyName: ec2-key-name
Scheduling:
  Scheduler: slurm
SlurmQueues:
```

```
- ...
Networking:
  SubnetIds:
    - subnet-34567890 # subnet with proxy
  AssignPublicIp: false
  Proxy:
    HttpProxyAddress: http://proxy-address:port
```

當 [Scheduling/SlurmQueuesNetworking](#)/設定 [AssignPublicIp](#) 為 `false`，必須正確設定子網路，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 與 AWS Batch 調度程序

當您用 `awsbatch` 作排程器類型時，AWS ParallelCluster 會建立 AWS Batch 受管理的運算環境。該 AWS Batch 環境管理 Amazon Elastic Container Service (Amazon ECS) 容器實例。這些執行個體會在 [AwsBatchQueues/Networking/SubnetIds](#) 參數中設定的子網路中啟動。AWS Batch 為了正常運作，Amazon ECS 容器執行個體需要外部網路存取才能與 Amazon ECS 服務端點進行通訊。這轉換成以下案例：

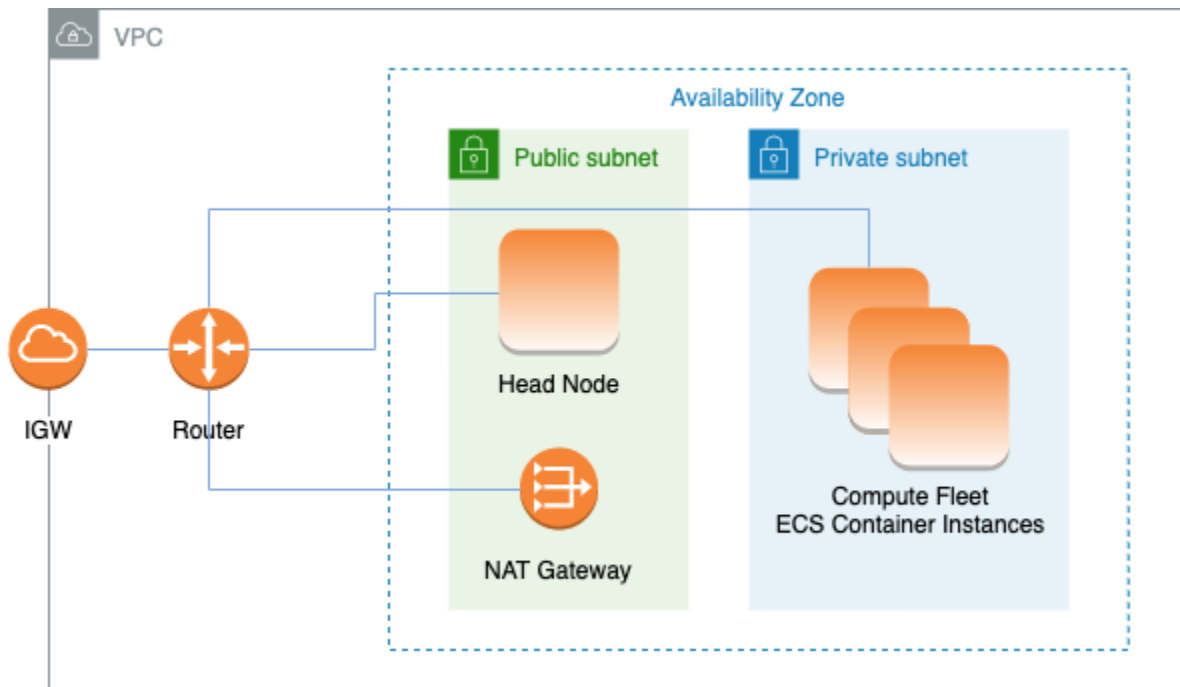
- 為佇列指定的子網路 ID 會使用 [NAT 閘道](#) 來存取網際網路。我們建議使用這種方法。
- 佇列子網路中啟動的執行個體具有公用 IP 位址，而且可以透過網際網路 Internet Gateway 路。

此外，如果您對多節點 `parallel` 作業感興趣（來自文 [AWS Batch 檔](#)）：

AWS Batch 多節點 `parallel` 任務使用 Amazon ECS `awsvpc` 網路模式。這為您的多節點 `parallel` 任務容器提供與 Amazon EC2 執行個體相同的聯網屬性。每個多節點平行任務容器皆會取得自己的彈性網路界面、主要私有 IP 地址及內部 DNS 主機名稱。網路界面建立在與其主機運算資源相同的 Amazon VPC 子網路中。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 任務聯網時，`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 地址的彈性網路界面。若要存取網際網路，使用 Amazon EC2 啟動類型的任務必須在設定為使用 NAT 閘道的私有子網路中啟動。

您必須設定 [NAT 閘道](#)，才能讓叢集執行多節點 `parallel` 工作。



所有先前的配置和注意事項也適用於 AWS Batch。以下是 AWS Batch 網路組態的範例。

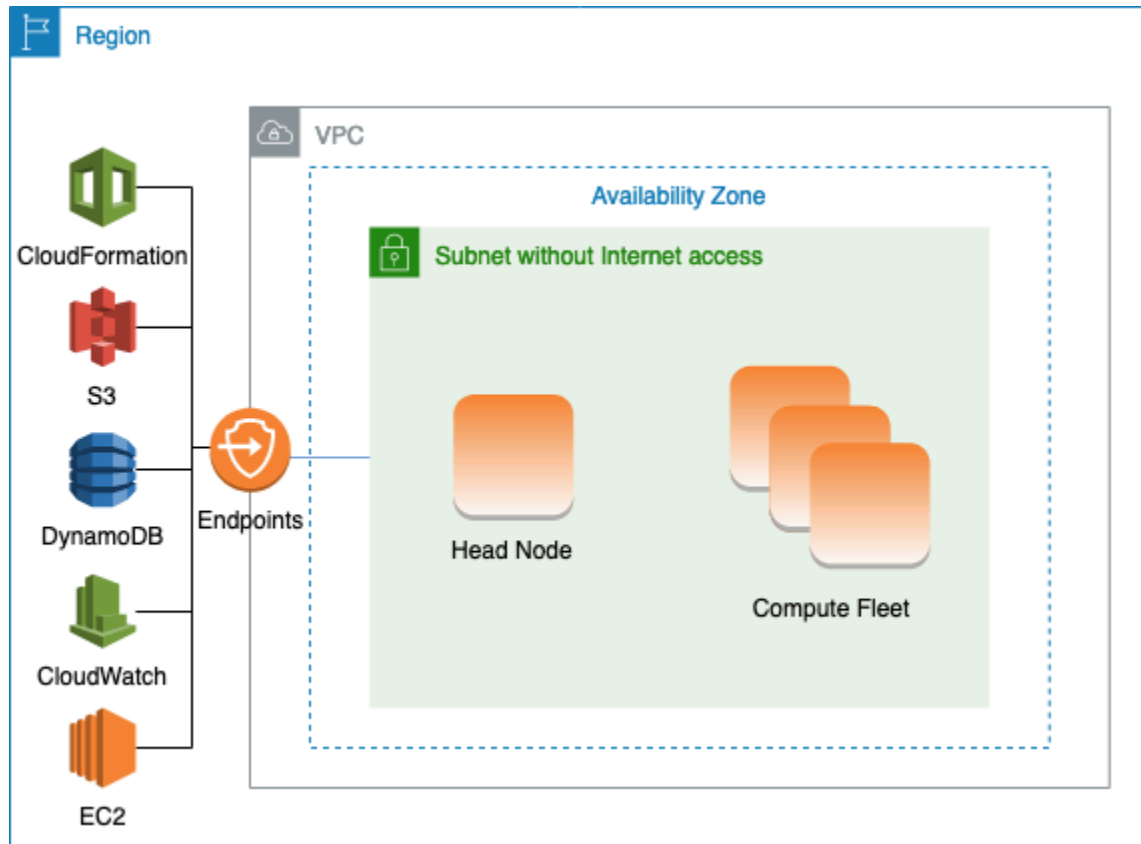
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway, NAT gateway or proxy
    #ElasticIp: true | false | eip-12345678
    #Proxy:
      #HttpProxyAddress: http://proxy-address:port
  Ssh:
    KeyName: ec2-key-name
  Scheduling:
    Scheduler: awsbatch
  AwsBatchQueues:
    - ...
      Networking:
        SubnetIds:
          - subnet-23456789 # subnet with internet gateway, NAT gateway or proxy
          #AssignPublicIp: true | false
```

在 [Scheduling/AwsBatchQueues/Networking](#) 區段中，[SubnetIds](#) 是清單類型，但目前只支援一個子網路。

如需詳細資訊，請參閱下列主題：

- [AWS Batch 受管理運算環境](#)
- [AWS Batch 多節點 parallel 工作](#)
- [使用 awsvpc 網路模式的 Amazon ECS 任務聯網](#)

## AWS ParallelCluster 在沒有互聯網訪問的單個子網中



沒有網際網路存取權的子網路不允許輸入或輸出連線至網際網路。此 AWS ParallelCluster 組態可協助與安全性相關的客戶進一步加強其 AWS ParallelCluster 資源的安全性。AWS ParallelCluster 節點 AWS ParallelCluster 是從 AMI 建立，其中包含執行叢集無法存取網際網路時所需的所有軟體。如此一來，就 AWS ParallelCluster 可以使用無法存取網際網路的節點來建立和管理叢集。

在本節中，您將瞭解如何設定叢集。您也會瞭解在沒有網際網路存取的情況下執行叢集的限制

### 設定 VPC 端點

若要確保叢集的正常運作，叢集節點必須能夠與許多 AWS 服務互動。

建立並設定下列 [VPC 端點](#)，以便叢集節點可以在沒有網際網路存取的情況下與 AWS 服務互動：



## Commercial and AWS GovCloud (US) partitions

服務	服務名稱	Type
Amazon CloudWatch	COM. 亞馬遜。 #####. ##	介面
AWS CloudFormation	COM. 亞馬遜。 ## ID. ###	介面
Amazon EC2	COM. 亞馬遜。 ##### .ec2	介面
Amazon S3	COM. 亞馬遜。 ##### .s3	閘道
Amazon DynamoDB	COM. 亞馬遜。 #####. ##	閘道
AWS Secrets Manager**	COM. 亞馬遜。 ## ID. ### #	介面

## China partition

服務	服務名稱	Type
Amazon CloudWatch	COM. 亞馬遜。 #####. ##	介面
AWS CloudFormation	阿馬遜. 阿馬遜. ## ID. ###	介面
Amazon EC2	阿馬遜. 阿馬遜. ##### .ec2	介面
Amazon S3	COM. 亞馬遜。 ##### .s3	閘道
Amazon DynamoDB	COM. 亞馬遜。 #####. ##	閘道
AWS Secrets Manager**	COM. 亞馬遜。 ## ID. ### #	介面

\*\* 只有在啟用時[DirectoryService](#)才需要此端點，否則為選用端點。

VPC 中的所有執行個體都必須具有適當的安全群組，才能與端點通訊。您可以透過將安全性群組新增至組態[AdditionalSecurityGroups](#)下[HeadNode](#)和組[SlurmQueues](#)態[AdditionalSecurityGroups](#)下方來執行此操作。例如，如果在未明確指定

安全群組的情況下建立 VPC 端點，則預設安全群組會與端點相關聯。透過將預設安全性群組新增至 `AdditionalSecurityGroups`，即可啟用叢集與端點之間的通訊。

### Note

當您使用 IAM 政策限制對虛擬私人雲端端點的存取時，必須將以下內容新增到 Amazon S3 VPC 端點：

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

停用路由 53 並使用 Amazon EC2 主機名稱

建立 Slurm 叢集時，AWS ParallelCluster 會建立私有 Route 53 託管區域，用於解析自訂計算節點主機名稱，例如 `{queue_name}-{st|dy}-{compute_resource}-{N}`。由於 Route 53 不支援 VPC 端點，因此必須停用此功能。此外，AWS ParallelCluster 必須設定為使用預設的 Amazon EC2 主機名稱，例如 `ip-1-2-3-4`。將下列設定套用至叢集配置：

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

### Warning

對於使用 [SlurmSettings/Dns](#)/建立 [DisableManagedDns](#) 並 [UseEc2Hostnames](#) 設定為的叢集 `true`，DNS Slurm NodeName 無法解析。請 `SlurmNodeHostName` 改用。

**Note**

此註釋與 AWS ParallelCluster 版本 3.3.0 開始無關。

對於 3.3.0 之前的 AWS ParallelCluster 支援版本：

設定 `UseEc2Hostnames` 為 `true`，會使用 AWS ParallelCluster `prolog` 和 `epilog` 指令碼設定 Slurm 組態檔案：

- `prolog` 執行以在分配每個工作時，將節點資訊新增至運算節點 `/etc/hosts` 上。
- `epilog` 運行以清除寫入的內容 `prolog`。

若要新增自訂 `prolog` 或 `epilog` 指令碼，請將它們分別新增至 `/opt/slurm/etc/pcluster/prolog.d/` 或 `/opt/slurm/etc/pcluster/epilog.d/` 資料夾。

## 叢集配置

瞭解如何設定叢集在沒有網際網路連線的子網路中執行。

此架構的組態需要下列設定：

```
# Note that all values are only provided as examples
...
HeadNode:
  ...
  Networking:
    SubnetId: subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
    endpoints
    AdditionalSecurityGroups:
      - sg-abcdef01234567890 # optional, the security group that enables the
      communication between the cluster and the VPC endpoints
  Scheduling:
    Scheduler: Slurm # Cluster in a subnet without internet access is supported only when
    the scheduler is Slurm.
  SlurmSettings:
    Dns:
      DisableManagedDns: true
      UseEc2Hostnames: true
  SlurmQueues:
    - ...
      Networking:
        SubnetIds:
```

```
- subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
endpoints attached
AdditionalSecurityGroups:
- sg-1abcdef01234567890 # optional, the security group that enables the
communication between the cluster and the VPC endpoints
```

- [SubnetId\(s\)](#) : 沒有網際網路存取的字網路。

若要啟用 AWS ParallelCluster 和 AWS 服務之間的通訊，子網路的 VPC 必須已連接 VPC 端點。建立叢集之前，請確認子網路中的[自動指派公用 IPv4 位址已停用](#)，以確保 pcluster 命令可以存取叢集。

- [AdditionalSecurityGroups](#) : 啟用叢集與 VPC 端點之間通訊的安全性群組。

選用：

- 如果在未明確指定安全群組的情況下建立 VPC 端點，則會關聯 VPC 的預設安全性群組。因此，請將預設安全性群組提供給 AdditionalSecurityGroups。
- 如果在建立叢集和/或 VPC 端點時使用了自訂安全群組，AdditionalSecurityGroups 則只要自訂安全群組啟用叢集與 VPC 端點之間的通訊，就不需要。
- [Scheduler](#) : 叢集排程器。

slurm 是唯一有效的值。只有 Slurm 排程器支援沒有網際網路存取權的字網路中的叢集。

- [SlurmSettings](#) : 設 Slurm 置。

請參閱上一節停用路由 53 並使用 Amazon EC2 主機名稱。

## 限制

- 透過 SSH 或 NICE DCV 連線至頭節點：連線至叢集時，請確定連線的用戶端可透過其私有 IP 位址到達叢集的頭節點。如果用戶端與頭節點不在相同的 VPC 中，請在 VPC 的公有子網路中使用 Proxy 執行個體。此需求同時適用於 SSH 和 DCV 連線。如果子網路沒有網際網路存取權，則無法存取頭節點的公用 IP。pcluster ssh 和指 dcv-connect 命令會使用公用 IP (如果存在) 或私有 IP。建立叢集之前，請確認子網路中的[自動指派公用 IPv4 位址已停用](#)，以確保 pcluster 命令可以存取叢集。

下列範例顯示如何連線至叢集標頭節點中執行的 DCV 工作階段。您可以透過代理 Amazon EC2 執行個體連線。執行個體可做為您電腦的 NICE DCV 伺服器，以及私有子網路中頭節點的用戶端。

透過公有子網路中的 Proxy 執行個體透過 DCV Connect 線：

1. 在公有子網路中建立 Amazon EC2 執行個體，該子網路與叢集子網路位於相同的 VPC 中。
  2. 確保 NICE DCV 用戶端和伺服器已安裝在您的 Amazon EC2 執行個體上。
  3. 將 AWS ParallelCluster 使用者政策附加到代理 Amazon EC2 執行個體。如需詳細資訊，請參閱 [AWS ParallelCluster pcluster 使用者策略範例](#)。
  4. AWS ParallelCluster 在代理 Amazon EC2 實例上安裝。
  5. 透過 DCV Connect 到代理伺服器亞馬遜 EC2 執行個體。
  6. 使用 Proxy 執行個體上的 `pcluster dcv-connect` 指令，在沒有網際網路存取權的情況下連線至子網路內的叢集。
- 與其他 AWS 服務互動：上面僅列出嚴格要求的服務。AWS ParallelCluster 如果您的叢集必須與其他服務互動，請建立對應的 VPC 端點。

## 登入節點

從 3.7.0 版開始，AWS ParallelCluster 叢集管理員可以佈建登入節點，這些節點可用來提供使用者執行工作與直接存取叢集標頭節點的存取權。具有適當權限的群集用戶可以使用 Active Directory 或他們的 ssh 憑據登錄，提交和管理他們的作業。因此，可以改善叢集管理，並且可以將耗盡管理叢集所需之主節點資源的機會降 Slurm 到最低。登入的使用者也可以存取掛載在登入節點上之叢集的所有共用儲存裝置。如果需要停止登錄節點，則用戶將通過他們正在使用的活動 shell 會話提前通知用戶。

登入節點指定為集區，其中集區會定義具有相同資源配置的登入節點群組。集區中的所有登入節點都設定為 [網路負載平衡器](#) 的一部分，可讓您以循環配置方式跨登入節點分配工作階段。本實作允許指定一個包含多個登入節點的登入節點集區。

### 安全性

登錄節點會繼承頭節點的 AllowedIPs 設置 [AllowedIps](#)。透過這種方式，叢集管理員可以指定允許 SSH 連線的來源 CIDR 或首碼清單，以限制叢集的安全性狀況。

在目前的實作中，啟用登入節點時，不會自動限制對 head 節點的存取。如果需要，叢集管理員可以限制此存取權限，使用標準 Linux 指令更新 head 節點 ssh 組態。這也可以通過使用 ParallelCluster YAML 文件的頭節點部分中的 `AdditionalSecurityGroups` 設置來拒絕未經授權的用戶的連接，在頭節點上指定自定義安全組來實現。

### 聯網

登入節點會以單一連線位址佈建至為登入節點集區設定的網路負載平衡器。位址的連線設定是以登入節點集區組態中指定的子網路類型為基礎。

- 如果子網路是私人的，則該位址將是私有的，而且為了授與登入節點的存取權，叢集管理員必須佈建防禦主機。
- 如果子網路是公用的，則該位址將是公用的

所有連線要求均由 Network Load Balancer 使用循環配置資源路由來管理。

## 儲存

在叢集上設定的所有共用儲存裝置 ( ParallelCluster 包括受管理的儲存裝置) 都會掛載在所有登入節點上。

## 擷取登入節點資訊

若要擷取為存取登入節點而佈建的單一連線位址，叢集管理員可以執行[describe-cluster](#)命令。該命令還將提供有關登錄節點狀態的更多信息。

登入節點是一種支援的新節點類型 ParallelCluster ，可在查詢特定節點類型的狀態時使用指[describe-cluster-instances](#)令指定。

登入節點集區的單一連線位址可用性並不會阻止直接存取特定登入節點。但是，不建議使用直接連接以避免來自 ssh 客戶端的警告。ssh 用戶端會在本機儲存每個目標位址的主機識別碼。由於主機標識符是每個集區特定的，因此使用不同的 IP 和/或單個連接地址可能具有與不同目標地址相關聯的相同主機標識符：這可能會導致 ssh 客戶端發出警告，因為同一個主機標識符與多個目標相關聯。

## 伊姆斯屬性

登入節點的 IMDS ( 和執行個體設定檔登入資料 ) 的存取權限僅限於根使用者、叢集管理使用者 ( pc-cluster-admin預設情況下 ) 和作業系統特定的預設使用者 ( ec2-user在 Amazon Linux 2 和 Ubuntu 18.04 ubuntu 上 RedHat , 則在 CentOS 7 centos 上 ) 。

若要限制 IMDS 存取，請 AWS ParallelCluster 管理的iptables鏈結。

### Note

iptables或ip6tables規則的任何自訂都可能干擾用來限制登入節點上 IMDS 存取的機制。另請參閱。[Imds property setting](#)

## 登入節點的生命

目前沒有專用指令可停止及啟動集區中的登入節點。為了停止集區中的登入節點，叢集管理員必須更新登入節點 (Count: 0) 計數上指定 0 的叢集配置，然後執行指令 `pcluster.update-cluster-v3`。

### Note

已登入的使用者會收到關於特定執行個體終止的通知，以及相關的優惠期間。在寬鬆期間，除了叢集預設使用者的連線外，不允許任何新的連線。叢集管理員可從頭節點或編輯檔案的登入節點自訂顯示的訊息 `/opt/parallelcluster/shared_login_nodes/loginmgtd_config.json`。

為了啟動登入節點集區，叢集管理員必須還原叢集配置中先前的 Count 值，然後執行 `update-cluster` 指令。

執行登入節點集區所需的權限

若要管理登入節點集區，叢集管理員必須具備下列其他權限：

```
- Action:
  - autoscaling:DeleteAutoScalingGroup
  - autoscaling:DeleteLifecycleHook
  - autoscaling:Describe*
  - autoscaling:PutLifecycleHook
  - autoscaling:UpdateAutoScalingGroup
  - elasticloadbalancing:CreateListener
  - elasticloadbalancing:CreateTargetGroup
  - elasticloadbalancing>DeleteListener
  - elasticloadbalancing>DeleteLoadBalancer
  - elasticloadbalancing>DeleteTargetGroup
  - elasticloadbalancing:Describe*
  - elasticloadbalancing:ModifyLoadBalancerAttributes
Resource: '*'
Condition:
  ForAllValues:StringEquals:
    aws:TagKeys: [ "parallelcluster:cluster-name" ]
- Action:
  - autoscaling:CreateAutoScalingGroup
  - elasticloadbalancing:AddTags
  - elasticloadbalancing:CreateLoadBalancer
Resource: '*'
```

Effect: Allow

## 自訂引導動作

如果您定義 [HeadNode/CustomActions/OnNodeStart](#) 組態設定，請在節點啟動後立即 AWS ParallelCluster 執行任意程式碼。如果您定義 [HeadNode/CustomActions/OnNodeConfigured](#) 配置設置，則在節點配置正確完成後 AWS ParallelCluster 運行代碼。

從 AWS ParallelCluster 版本 3.4.0 開始，如果您定義 [HeadNode/CustomActions/OnNodeUpdated](#) 配置設置，則可以在頭節點更新後運行代碼。

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。程式碼會以叢集作業系統支援的任何指令碼語言執行，root 而且可以使用任何指令碼語言執行。通常代碼是在 Bash 或 Python 中。

### Note

從 AWS ParallelCluster 版本 3.7.0 開始，叢集 [Imds/ImdsSupport](#) 設定預設值為 v2.0。當您建立要升級至 3.7.0 版及更新版本的新叢集時，請將自訂啟動程序動作指令碼更新為與 IMDSv2 相容，或在叢集配置檔 v1.0 中 [ImdsSupport](#) 將 [Imds/](#) 設定為。

### Warning

您必須負責設定 [共用職責模型](#) 中所述的自訂指令集與引數。確認您的自訂啟動程序指令碼和引數來自您信任的來源，可完整存取叢集節點。

### Warning

AWS ParallelCluster 不支持使用通過 `/etc/parallelcluster/cfnconfig` 文件提供的內部變量。此檔案可能會在 future 版本中移除。

`OnNodeStart` 在啟動任何節點部署啟動程序動作 (例如設定 NAT、Amazon Elastic Block Store (Amazon EBS) 或排程器之前，都會呼叫動作。`OnNodeStart` 引導操作可能包括修改存儲，添加額外的用戶和添加軟件包。



**Note**

如果您為叢集設定 [DirectoryServiceHeadNodeCustomActions//OnNodeStart](#) 指令碼，請在執行 AWS ParallelCluster 指令碼之前設定 DirectoryServiceSsd 並重新啟動。OnNodeStart

OnNodeConfigured 動作會在節點啟動程序完成後呼叫。OnNodeConfigured 動作會在將執行個體視為完全設定並完成之前發生的最後一個動作。某些 OnNodeConfigured 動作包括變更排程器設定、修改儲存區和修改套件。您可以通過在配置過程中指定參數來將參數傳遞給腳本。

OnNodeUpdated 在頭節點更新完成，並且排程器和共用儲存裝置與最新的叢集配置變更一致之後，會呼叫動作。

當 OnNodeConfigured 自訂動作成功 OnNodeStart 或自訂動作成功時，會以結束代碼零 (0) 表示成功。任何其他結束代碼表示執行個體啟動程序失敗。

當 OnNodeUpdated 自定義操作成功時，成功的信號與退出代碼零 (0)。任何其他結束代碼都表示更新失敗。

**Note**

如果您進行設定 [OnNodeUpdated](#)，則必須在更新失敗時手 OnNodeUpdated 動將動作還原到先前的狀態。

如果自 OnNodeUpdated 訂動作失敗，更新會復原至先前的狀態。不過，OnNodeUpdated 動作只會在更新時執行，而不會在堆疊回復時執行。

您可以在 / 和 [HeadNode//CustomActionsCustomActions](#) 配置部分為頭節點和每個隊列指定不同的腳本。[SchedulingSlurmQueues OnNodeUpdated](#) 只能在 HeadNode 區段中設定。

**Note**

在 3.0 AWS ParallelCluster 版之前，無法為頭節點和計算節點指定不同的腳本。請參閱 [從 AWS ParallelCluster 2.x 移動到 3.x](#)。

**主題**

- [組態](#)

- [引數](#)
- [包含自訂引導動作的範例叢集](#)
- [更新 IMDSv2 的自訂啟動程序指令碼範例](#)
- [更新 IMDSv1 組態的範例](#)

## 組態

以下配置設置用於定義 [HeadNodeCustomActions//OnNodeStartOnNodeConfigured&OnNodeUpdated](#) 和 [Scheduling/CustomActions/OnNodeStart& OnNodeConfigured](#) 操作和參數。

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URL. This is run before any of the bootstrap scripts are run
      Script: s3://bucket-name/on-node-start.sh
      Args:
        - arg1
    OnNodeConfigured:
      # Script URL. This is run after all the bootstrap scripts are run
      Script: s3://bucket-name/on-node-configured.sh
      Args:
        - arg1
    OnNodeUpdated:
      # Script URL. This is run after the head node update is completed.
      Script: s3://bucket-name/on-node-updated.sh
      Args:
        - arg1
  # Bucket permissions
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false
  Scheduling:
    Scheduler: slurm
    [...]
  SlurmQueues:
    - Name: queue1
      [...]
    CustomActions:
```

```
OnNodeStart:
  Script: s3://bucket-name/on-node-start.sh
  Args:
    - arg1
OnNodeConfigured:
  Script: s3://bucket-name/on-node-configured.sh
  Args:
    - arg1
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false
```

使用Sequence設置 ( 在 AWS ParallelCluster 版本 3.6.0 中添加 ) :

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, before any of the bootstrap scripts are run.
      Sequence:
        - Script: s3://bucket-name/on-node-start1.sh
          Args:
            - arg1
        - Script: s3://bucket-name/on-node-start2.sh
          Args:
            - arg1
      [...]
    OnNodeConfigured:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, after all the bootstrap scripts are run.
      Sequence:
        - Script: s3://bucket-name/on-node-configured1.sh
          Args:
            - arg1
        - Script: s3://bucket-name/on-node-configured2.sh
          Args:
            - arg1
      [...]
    OnNodeUpdated:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, after the head node update is completed.
```

```
Sequence:
  - Script: s3://bucket-name/on-node-updated1.sh
  Args:
    - arg1
  - Script: s3://bucket-name/on-node-updated2.sh
  Args:
    - arg1
  [...]
# Bucket permissions
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false
Scheduling:
  Scheduler: slurm
  [...]
  SlurmQueues:
    - Name: queue1
    [...]
  CustomActions:
    OnNodeStart:
      # Script URLs. The scripts are run in the same order as listed in the
      configuration, before any of the bootstrap scripts are run
      Sequence:
        - Script: s3://bucket-name/on-node-start1.sh
        Args:
          - arg1
        - Script: s3://bucket-name/on-node-start2.sh
        Args:
          - arg1
        [...]
    OnNodeConfigured:
      # Script URLs. The scripts are run in the same order as listed in the
      configuration, after all the bootstrap scripts are run
      Sequence:
        - Script: s3://bucket-name/on-node-configured1.sh
        Args:
          - arg1
        - Script: s3://bucket-name/on-node-configured2.sh
        Args:
          - arg1
        [...]
  Iam:
    S3Access:
```

```
- BucketName: bucket_name
  EnableWriteAccess: false
```

從 AWS ParallelCluster 版本 3.6.0 開始新增此 Sequence 設定。當您指定時 Sequence，您可以列出自訂動作的多個指令碼。AWS ParallelCluster 繼續支援使用單一指令碼設定自訂動作，但不包含 Sequence。

AWS ParallelCluster 不支持包括單個腳本和相同 Sequence 的自定義操作。例如，如 AWS ParallelCluster 果您指定下列組態，就會失敗。

```
[...]
CustomActions:
  OnNodeStart:
    # Script URL. This is run before any of the bootstrap scripts are run
    Script: s3://bucket-name/on-node-start.sh
    Args:
      - arg1
    # Script URLs. The scripts are run in the same order as listed in the
    configuration, before any of the bootstrap scripts are run.
    Sequence:
      - Script: s3://bucket-name/on-node-start1.sh
        Args:
          - arg1
      - Script: s3://bucket-name/on-node-start2.sh
        Args:
          - arg1
[...]
```

## 引數

### Note

在 AWS ParallelCluster 2.x 中，\$1 參數是一個保留的，用於存儲自定義腳本的 URL。如果要重新使用為 AWS ParallelCluster 2.x 和 AWS ParallelCluster 3.x 創建的自定義引導腳本，則需要通過考慮參數的移位來調整它們。請參閱 [從 AWS ParallelCluster 2.x 移動到 3.x](#)。

## 包含自訂引導動作的範例叢集

下列步驟會建立在設定節點後執行的簡單指令碼，以便在叢集的節點中安裝 R, curl 和 wget 套件。

## 1. 建立指令碼。

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

2. 將具有正確許可的指令碼上傳到 Amazon S3。如果公共讀取權限不適合您，請使用 [HeadNode/Iam/S3Access](#) 和 [Scheduling/SlurmQueues](#) 配置部分。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

如果指令碼是在 Windows 上編輯的，則必須先將行尾從 CRLF 變更為 LF，才能將指令碼上傳到 Amazon S3。

3. 更新組 AWS ParallelCluster 態以包含新 OnNodeConfigured 動作。

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

如果值區沒有公開讀取權限，請 s3 做為 URL 通訊協定使用。

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

#### 4. 啟動叢集。

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

#### 5. 驗證輸出。

- 如果您已將自訂動作新增至HeadNode組態，請登入 head 節點，並/var/log/cfn-init.log透過執行下列命令來檢查位於的cfn-init.log檔案：

```
$ less /var/log/cfn-init.log  
2021-09-03 10:43:54,588 [DEBUG] Command run  
postinstall output: The script has 3 arguments  
arg: R  
arg: curl  
arg: wget  
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper  
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version  
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version  
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version  
Nothing to do
```

- 如果您已將自訂動作新增至SlurmQueues設定，請勾選cloud-init.log位/var/log/cloud-init.log於計算節點中的。用 CloudWatch 於檢視這些記錄檔。

您可以在 Amazon CloudWatch 主控台中檢視這兩個日誌。如需詳細資訊，請參閱 [與 Amazon Amazon CloudWatch 的整合](#)。

## 更新 IMDSv2 的自訂啟動程序指令碼範例

在下列範例中，我們更新了與 IMDSv1 搭配使用的自訂啟動程序動作指令碼，以搭配 IMDSv2 使用。IMDSv1 指令碼會擷取亞馬遜 EC2 執行個體 AMI ID 中繼資料。

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

以下顯示修改為與 IMDSv2 相容的自訂啟動程序動作指令碼。

```
#!/bin/bash
```

```
AMI_ID=$(TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

如需詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的[擷取執行個體中繼資料](#)。

## 更新 IMDSv1 組態的範例

以下是使用 3.7.0 及更 AWS ParallelCluster 舊版本時支援 IMDSv1 的叢集配置範例。

```
Region: us-east-1
Imds:
  ImdsSupport: v1.0
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh
    KeyName: key-name
  CustomActions:
    OnNodeConfigured:
      Script: Script-path
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CustomActions:
        OnNodeConfigured:
          Script: Script-path
  ComputeResources:
    - Name: t2micro
      Instances:
        - InstanceType: t2.micro
      MinCount: 11
  Networking:
    SubnetIds:
      - subnet-abcdef01234567890
```



## 使用 Amazon S3

您可以透過AWS ParallelCluster組態中AWS ParallelCluster的//[S3Access](#)和[HeadNode///IamSchedulingSlurmQueues- NameIam/S3Access](#)參數設定對 Amazon S3 的存取權限。

### 範例

下列範例會設定中所有物件的唯讀存取權，以`firstbucket/read_only/`及對中所有物件的讀取/寫入存取權`secondbucket/read_and_write/`。

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

下一個範例會設定對帳戶中任何值區 (\*) 資料夾`read_only/`中所有物件的唯讀存取權。

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

最後一個範例會設定對帳戶中所有值區和物件的 `read_only` 存取權。

```
...
HeadNode:
```

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

## 使用 競價型執行個體

AWS ParallelCluster 如果您已SPOT在叢集配置檔中將 [SlurmQueues/CapacityType](#) 或 [AwsBatchQueues](#)/設定 [CapacityType](#) 為，則會使用 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。這可能有助於利用 Spot 執行個體中斷通知，該通知在 Amazon EC2 必須停止或終止 Spot 執行個體之前提供兩分鐘的警告。[如需詳細資訊，請參閱 Amazon EC2 使用者指南中的競價型執行個體中斷](#)。要了解如何使 [AwsBatchQueues](#) 用 Spot 執行個體，請參閱 AWS Batch 使用者指南中的 [計算資源](#)。

AWS ParallelCluster 設定的排程器會將任務指派給具有 Spot 執行個體之佇列中的運算資源，方式與將任務指派給具有隨需執行個體之佇列中的運算資源相同

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](#)。

以下各節說明 Spot 執行個體在使用時可能會中斷的三種案例 [SlurmQueues](#)。

### 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，如果排程器佇列有擱置的工作需要額外執行個體，或作用中執行個體的數目低於 [SlurmQueues/ComputeResources/](#)，則 AWS ParallelCluster 嘗試取代執行個體 [MinCount](#)。如果 AWS ParallelCluster 無法佈建新執行個體，則會定期重複新執行個體的要求。

### 案例 2：執行單一節點任務的 Spot 執行個體遭到中斷

工作失敗，狀態碼為 `NODE_FAIL`，且會重新計算工作 (除 `--no-requeue` 非在送出工作時指定)。如果節點是靜態節點，則會取代它。如果節點是動態節點，則會終止並重設節點。如需有關 `sbatch` (包括參數) 的詳細資訊，請 `--no-requeue` 參閱 Slurm 文件 [sbatch](#) 中的。

## 案例 3：執行多節點任務的 Spot 執行個體遭到中斷

工作失敗，狀態碼為 `NODE_FAIL`，且會重新計算工作 (除非 `--no-requeue` 在送出工作時指定)。如果節點是靜態節點，則會取代它。如果節點是動態節點，則會終止並重設節點。執行已終止工作的其他節點可能會配置給其他擱置中的工作，或在設定的 [SlurmSettings/ScaledownIdleTime](#) 時間過後縮減。

如需競價型執行個體的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [競價型執行個體](#)。

## 排程器支援 AWS ParallelCluster

### 排程器支援 AWS ParallelCluster

AWS ParallelCluster 支援 Slurm 和 AWS Batch 調度程序，使用該 [Scheduler](#) 設置進行設置。

#### 主題

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

## Slurm Workload Manager (**slurm**)

### 叢集容量大小和更新

叢集的容量是由叢集可擴充的運算節點數量所定義。運算節點由組 AWS ParallelCluster 態中運算資源中定義的 Amazon EC2 執行個體提供支援 ([Scheduling/SlurmQueues/ComputeResources](#))，並組織成佇列 ([Scheduling/SlurmQueues](#))，以對應 1:1 到 Slurm 分區。

在計算資源中，您可以設定叢集中必須始終保持執行的最小計算節點 (執行個體 [MinCount](#)) 數目，以及運算資源可擴充至 ([MaxCount](#)) 的執行個體數目上限。

在叢集建立時或叢集更新時，[MinCount](#) 針對叢集中定義的每個運算資源 ([Scheduling/SlurmQueues/ComputeResources](#)) AWS ParallelCluster 啟動所設定數量的 Amazon EC2 執行個體。為了涵蓋叢集中運算資源的最小節點而啟動的執行個體稱為靜態節點。一旦啟動，靜態節點就應該在叢集中是永久性的，除非發生特定的事件或條件，否則它們不會被系統終止。例如，此類事件包括 Slurm 或 Amazon EC2 運作狀態檢查失敗，以及將 Slurm 節點狀態變更為「排出」或「關閉」。

Amazon EC2 執行個體的範圍 1 為 'MaxCount - MinCount' (MaxCount 減號 MinCount)，隨需啟動以處理叢集增加的負載) 稱為動態節點。它們的性質是短暫的，它們被啟動以提供待處理的工作，

並且一旦它們Scheduling/SlurmSettings/[ScaledownIdleTime](#)在叢集配置中定義的一段時間內保持閒置狀態 ( 預設值：10 分鐘 )，就會終止。

靜態節點和動態節點符合下列命名結構描述：

- 靜態節點<Queue/Name>-st-<ComputeResource/Name>-<num>，其中 <num> = 1..ComputeResource/MinCount
- 動態節點<Queue/Name>-dy-<ComputeResource/Name>-<num>，其中 <num> = 1..(ComputeResource/MaxCount - ComputeResource/MinCount)

例如，給出以下 AWS ParallelCluster 配置：

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

下面的節點將被定義 Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

當計算資源具有時MinCount == MaxCount，所有對應的計算節點都將是靜態的，並且所有執行個體都會在叢集建立/更新時啟動，並保持正常運作。例如：

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
```

```
- Name: queue1
  ComputeResources:
    - Name: c5xlarge
      Instances:
        - InstanceType: c5.xlarge
      MinCount: 100
      MaxCount: 100
```

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

## 叢集容量更新

叢集容量的更新包括新增或移除佇列、計算資源或變更計算資源。MinCount/MaxCount從 AWS ParallelCluster 版本 3.9.0 開始，減少佇列的大小需要停止運算叢集或 [QueueUpdateStrategy](#) 設定為「終止」，才能進行叢集更新。在下列情況下，不需要停止運算叢集或設定 [QueueUpdateStrategy](#) 為「終止」：

- 將新佇列新增至排程/[SlurmQueues](#)
- 將新的計算資源新增 Scheduling/SlurmQueues/[ComputeResources](#) 至佇列
- 增加 [MaxCount](#) 運算資源
- 增加 MinCount 運算資源，並增加 MaxCount 至少相同數量的相同運算資源

## 考量與限制

本節旨在概述調整叢集容量大小時應考慮的任何重要因素、限制或限制。

- 從名稱 <Queue/Name>-\* 為靜態和動態的 Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues 所有運算節點移除佇列時，將從 Slurm 組態中移除，而對應的 Amazon EC2 執行個體也會終止。
- Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues>

[ComputeResources](#) ComputeResources 從佇列中移除運算資源時，所有具有靜態和動態名稱 `<Queue/Name>-*-<ComputeResource/Name>-*` 的運算節點都會從 Slurm 組態中移除，而對應的 Amazon EC2 執行個體也會終止。

當更改計算資源的 MinCount 參數時，我們可以區分兩種不同的場景，如果 MaxCount 保持等於 MinCount ( 僅靜態容量 )，以及 MaxCount 是否大於 MinCount ( 混合靜態和動態容量 )。

### 僅限靜態節點的容量變更

- 如果在增加 MinCount (和 MaxCount) 時  $\text{MinCount} == \text{MaxCount}$ ，將透過將靜態節點數量擴展到的新值來設定叢集，`MinCount<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` 且系統會繼續嘗試啟動 Amazon EC2 執行個體以滿足新的所需靜態容量。
- 如果  $\text{MinCount} == \text{MaxCount}$  在減少 MinCount (和 MaxCount) 數量 N 時，將透過移除最後 N 個靜態節點來設定叢集 `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`]，系統將終止對應的 Amazon EC2 執行個體。
- 初始狀態  $\text{MinCount} = \text{MaxCount} = 100$

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

- `-30` 在 MinCount 和更新 MaxCount:  $\text{MinCount} = \text{MaxCount} = 70$

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

### 混合節點的容量變更

如果  $\text{MinCount} < \text{MaxCount}$  增加 MinCount N 量 (假設 MaxCount 將保持不變)，則會將數量靜態節點擴充到新值  $\text{MinCount} (\text{old\_MinCount} + N)$  來設定叢集：`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` 系統將繼續嘗試啟動 Amazon EC2 執行個體以滿足新的必要靜態容量。此外，為了承受運算資源的 MaxCount 容量，透過移除最後 N 個

動態節點來更新叢集組態：`<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]`系統將終止對應的 Amazon EC2 執行個體。

- 初始狀態：MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 更新至 MinCount：MinCount = 130 (MaxCount = 150)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*    up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

如果在增加MinCount且數量相同MaxCount的 N 時  $\text{MinCount} < \text{MaxCount}$ ，將透過將數量靜態節點擴展到  $\text{MinCount} (\text{old\_MinCount} + N)$  的新值來設定叢集：`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>`系統將繼續嘗試啟動 Amazon EC2 執行個體以滿足新的所需靜態容量。此外，不會對動態節點的數量進行任何更改以兌現新節點

MaxCount 值。

- 初始狀態：MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 更新至 MinCount：MinCount = 130 (MaxCount = 180)

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  130    idle queue1-st-c5xlarge-[1-130]

```

如果  $\text{MinCount} < \text{MaxCount}$  在減少  $\text{MinCount}$   $N$  量時 (假設保持不變), 則  $\text{MaxCount}$  會透過移除最後  $N$  個靜態節點靜態節點來設定叢集 `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>`, 系統將終止對應的 Amazon EC2 執行個體。此外, 為了承受運算資源的  $\text{MaxCount}$  容量, 透過擴充動態節點的數量來更新叢集組態以填補空白。  $\text{MaxCount} - \text{new\_MinCount}$ : `<Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]` 在這種情況下, 由於這些節點是動態節點, 因此除非排程器在新節點上有待處理的工作, 否則不會啟動新的 Amazon EC2 執行個體。

- 初始狀態:  $\text{MinCount} = 100$ ;  $\text{MaxCount} = 150$

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]

```

- 在上的 -30 更新  $\text{MinCount}$  :  $\text{MinCount} = 70$  ( $\text{MaxCount} = 120$ )

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   80    idle~ queue1-dy-c5xlarge-[1-80]
queue1*   up    infinite   70    idle queue1-st-c5xlarge-[1-70]

```

如果  $\text{MinCount} < \text{MaxCount}$  在減少  $\text{MinCount}$  且數  $\text{MaxCount}$  量相同  $N$  時, 將透過移除最後  $N$  個靜態節點來設定叢集 `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>]`, 系統將終止對應的 Amazon EC2 執行個體。



此外，不會對動態節點的數量進行任何更改以兌現新MaxCount值。

- 初始狀態：MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 在上的 -30 更新 MinCount：MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   80    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite   70    idle queue1-st-c5xlarge-[1-70]
```

如果MinCount < MaxCount在減少 MaxCount N 量時 (假設MinCount將保持不變)，則會透過移除最後 N 個動態節點來設定叢集，<Queue/Name>-dy-<ComputeResource/Name>-<old\_MaxCount - N...<oldMaxCount>]而在執行的情況下，系統將終止對應的 Amazon EC2 執行個體。預期不會對靜態節點造成任何影響。

- 初始狀態：MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 在上的 -30 更新 MaxCount：MinCount = 100 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
```

```
queue1*      up    infinite    20  idle~ queue1-dy-c5xlarge-[1-20]
queue1*      up    infinite    100 idle queue1-st-c5xlarge-[1-100]
```

## 對工作的影響

在移除節點且 Amazon EC2 執行個體終止的所有情況下，除非沒有其他節點滿足任務要求，否則已在已移除節點上執行的 sbatch 任務都會重新排入佇列。在最後一種情況下，工作將失敗，狀態為 NODE\_FAIL 並從隊列中消失；如果是這種情況，則需要手動重新提交。

如果您打算執行叢集調整大小更新，則可以防止工作在計劃更新期間移除的節點中執行。這可以通過設置要在維護中刪除的節點來實現。請注意，在維護中設定節點不會影響最終已在節點中執行的工作。

假設使用計劃的叢集調整大小更新，您將刪除節點 `queueu-st-computeresource-[9-10]`。您可以使用下列命令建立 Slurm 保留區

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

這將建立在節點 `maint_for_update` 上命名的 Slurm 保留區 `queueu-st-computeresource-[9-10]`。從建立保留區的時間起，節點就不能再執行任何作業 `queueu-st-computeresource-[9-10]`。請注意，保留區不會阻止最終在節點上配置工作 `queueu-st-computeresource-[9-10]`。

叢集調整大小更新後，如果只在調整大小更新期間移除的節點上設定保 Slurm 留區，則維護保留區將會自動刪除。如果您在叢集調整大小更新之後仍然存在的節點上建立保 Slurm 留區，我們可能想要在執行調整大小更新之後移除節點上的維護保留區，方法是使用下列命令

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

[有關 Slurm 預訂的更多詳細信息，請在此處查看官方 ScheDMD 文檔。](#)

## 容量變更的叢集更新程序

排程器組態變更時，會在叢集更新程序期間執行下列步驟：

- 停止 AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- 從配置生成更新的 Slurm 分區配 AWS ParallelCluster 置

- 重新啟動slurmctld ( 通過 Chef 服務配方完成 )
- 檢查slurmctld狀態 (systemctl is-active --quiet slurmctld.service)
- 重新載入Slurm組 (scontrol reconfigure)
- 啟動 clustermgtd (supervisorctl start clustermgtd)

如需有關的資訊Slurm，請參閱 <https://slurm.schedmd.com>。如需下載項目，請參閱 <https://github.com/SchedMD/slurm/tags>。如需原始程式碼，請參閱 <https://github.com/SchedMD/slurm>。

AWS ParallelCluster 版本	支援的 Slurm 版本
3.9.2、3.9.3、3.10.0	23.11.7
3.9.0	23.11.4
3.8.0	23.02.7
3.7.2	23.02.6
3.7.1	23.02.5
3.7.0	23.02.4
3.6.0	23.02.2
3.5.0	22.05.8
3.4.0	22.05.7
3.3.0	22.05.5
3.1.4、3.1.5、3.2.0、3.2.1	21.08.8-2
3.1.2	21.08.6
3.1.1	21.08.5
3.0.0	20.11.8

## 主題

- [多個佇列的組態](#)
- [Slurm多隊列模式指南](#)
- [Slurm叢集保護模式](#)
- [Slurm叢集快速容量不足容錯移轉](#)
- [Slurm記憶體式排程](#)
- [使用 Slurm 進行多個實例類型分配](#)
- [動態節點的叢集擴展](#)
- [Slurm會計與 AWS ParallelCluster](#)
- [Slurm組態自訂](#)
- [Slurmprolog 與 epilog](#)
- [叢集容量大小和更新](#)

## 多個佇列的組態

### 多個佇列的組態

使用 AWS ParallelCluster 版本 3 時，您可以[SlurmQueues](#)在組態檔案中將設定[Scheduler](#)為slurm並指定多個佇列，以配置多個佇列。在此模式中，不同的執行個體類型會共存於組態檔[ComputeResources](#)區段中指定的運算節點中。[ComputeResources](#)具有不同例證類型的比例會根據需要向上或縮小[SlurmQueues](#)。

### 叢集佇列和計算資源配額

資源	配額
<a href="#">Slurm queues</a>	每個叢集 50 個佇列
<a href="#">Compute resources</a>	每個佇列 50 個運算資源 每個叢集 50 個運算資源

### 節點計數

佇列中[ComputeResources](#)的每個計算資源都必須具有唯一的[NameInstanceTypeMinCount](#)、[MaxCount](#)和[MinCount](#)。[MinCount](#)並[MaxCount](#)具有定義佇列中[ComputeResources](#)計算資源執行個體範圍的預設值。您也可以為和指定自己[MinCount](#)的值[MaxCount](#)。中的每個計算資源

都[ComputeResources](#)是由編號為 1 到值的靜態節點，以[MinCount](#)及從的值[MinCount](#)到的值編號的動態節點組成[MaxCount](#)。

## 範例組態

以下是叢集配置檔的「[排程](#)」區段範例。在此配置中有兩個名為queue1queue2和的隊列，每個隊列都具[ComputeResources](#)有指定的[MaxCount](#)。

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - InstanceType: c5.xlarge
          MaxCount: 5
          Name: c5xlarge
        - InstanceType: c4.xlarge
          MaxCount: 5
          Name: c4xlarge
    - Name: queue2
      ComputeResources:
        - InstanceType: c5.xlarge
          MaxCount: 5
          Name: c5xlarge
```

## 主機名稱

系統會動態指派啟動至運算叢集的執行個體。會針對每個節點產生主機名稱。默認情況下 AWS ParallelCluster 將使用以下格式的主機名稱：

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

- \$QUEUE是佇列的名稱。例如，如果[SlurmQueues](#)區段的項目[Name](#)設定為「queue-name」，則「\$QUEUE」就是「queue-name」。
- \$STATDYNst用於靜態節點或dy動態節點。
- \$COMPUTE\_RESOURCE是對應於此節點[Name](#)的[ComputeResources](#)計算資源。
- \$NODENUM是節點的編號。\$NODENUM動態節點的值介[MinCount](#)於一 (1) 與靜態節點的值之間，在一 (1) 和[MaxCount](#)之間。[MinCount](#)

從上面的示例配置文件中，來自queue1和計算資源的給定節點c5xlarge具有一個主機名：queue1-dy-c5xlarge-1。

主機名稱和完整網域名稱 (FQDN) 都是使用 Amazon Route 53 託管區域建立的。FQDN 是 `$HOSTNAME.$CLUSTERNAME.pcluster`，其中 `$CLUSTERNAME` 是叢集的名稱。

請注意，Slurm 節點名稱也會使用相同的格式。

使用者可以選擇使用支援運算節點的執行個體的預設 Amazon EC2 主機名稱，而不是使用所使用的預設主機名稱格式 `AWS ParallelCluster`。這可以通過將 [UseEc2Hostnames](#) 參數設置為 `true` 來完成。但是，Slurm 節點名稱將繼續使用預設 `AWS ParallelCluster` 格式。

## Slurm 多隊列模式指南

您可以在這裡了解如何 AWS ParallelCluster Slurm 管理佇列 (分割區) 節點，以及如何監視佇列和節點狀態。

### 概觀

擴展架構基於 Slurm 的 [雲計劃指南](#) 和省電插件。如需省電外掛程式的詳細資訊，請參閱 [省 Slurm 電指南](#)。在架構中，可能可供叢集使用的資源通常會在 Slurm 組態中預先定義為雲端節點。

### 雲節點生命週期

在其生命週期中，雲端節點會輸入以下幾個 (如果不是全部 `pow_up`) 狀態：`POWER_SAVING`、`POWER_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。以下清單詳細介紹了雲端節點生命週期中這些狀態的幾個方面。

- **POWER\_SAVING** 狀態中的節點會在中顯示 `~` 字尾 (例如 `idle~`) `sinfo`。在此狀態下，沒有 EC2 執行個體支援節點。但是，仍然 Slurm 可以將工作分配給節點。
- 轉換至 **POWER\_UP** 狀態的節點會在中顯示 `#` 字尾 (例如 `idle#`)。當將工作 Slurm 分配到 `POWER_UP` 狀態中的節點時，節點會自動轉換到 `POWER_SAVING` 狀態。

或者，您可以使用以下指令，以 `su root` 使用者身分手動將節點轉換為 `POWER_UP` 狀態：

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

在此階段，會 `ResumeProgram` 叫用 EC2 執行個體並啟動並設定，並將節點轉換為 `POWER_UP` 態。

- 目前可供使用的節點會在中顯示不含字尾 (例如 `idle`) `sinfo`。節點設定完成並加入叢集之後，就可以執行工作。在此階段，節點已正確配置並準備使用。

一般而言，我們建議 Amazon EC2 執行個體的數量與可用節點的數量相同。在大多數情況下，建立叢集之後，即可使用靜態節點。

- 轉換為**POWER\_DOWN**狀態的節點會在sinfo中顯示%字尾 (例如idle%)。動態節點會在之後自動進入**POWER\_DOWN**狀態[ScaledownIdleTime](#)。相反，在大多數情況下，靜態節點不會關閉電源。不過，您可以使用下列指令，以 su root 使用者身分手動將節點置於**POWER\_DOWN**狀態：

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

在此狀態下，與節點相關聯的執行個體會終止，且節點會設回**POWER\_SAVING**狀態，並可在之後使用[ScaledownIdleTime](#)。

[ScaledownIdleTime](#)設定會儲存至組Slurm態SuspendTimeout設定。

- 離線的節點會顯示在中，並帶有\*字尾 (例如down\*) sinfo。如果Slurm控制器無法連絡節點，或者靜態節點已停用且支援執行個體終止，則節點會離線。

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-efacompute1-[1-4]
efa        up    infinite   1     idle  efa-st-efacompute1-1
gpu        up    infinite   1     idle% gpu-dy-gpucompute1-1
gpu        up    infinite   9     idle~ gpu-dy-gpucompute1-[2-10]
ondemand   up    infinite   2     mix#  ondemand-dy-ondemandcompute1-[1-2]
ondemand   up    infinite   18    idle~ ondemand-dy-ondemandcompute1-
[3-10],ondemand-dy-ondemandcompute2-[1-10]
spot*      up    infinite   13    idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*      up    infinite   2     idle  spot-st-spotcompute2-[1-2]
```

spot-st-spotcompute2-[1-2]和efa-st-efacompute1-1節點已設定備份執行個體，且可供使用。ondemand-dy-ondemandcompute1-[1-2]節點處於狀**POWER\_UP**態，應該在幾分鐘內可用。gpu-dy-gpucompute1-1節點處於**POWER\_DOWN**狀態，並在之後轉換為**POWER\_SAVING**狀態[ScaledownIdleTime](#) ( 默認為 10 分鐘 )。

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

## 使用可用節點

可用節點由 Amazon EC2 執行個體提供支援。根據預設，節點名稱可用來直接透過 SSH 存取執行個體 (例如 `ssh efa-st-efacompute1-1`)。您可以使用以下指令擷取執行個體的私有 IP 位址：

```
$ scontrol show nodes nodename
```

檢查傳回 `NodeAddr` 欄位中的 IP 位址。

對於無法使用的節點，`NodeAddr` 欄位不應指向執行中的 Amazon EC2 執行個體。相反，它應該是相同的節點名稱。

## Job 狀態和提交

在大多數情況下提交的工作會立即分配給系統中的節點，或者如果所有節點都已配置，則會置於擱置中。

如果為工作配置的節點包含 `POWER_SAVING` 狀態中的任何節點，則工作會以 `CF`、或 `CONFIGURING` 狀態開始。此時，工作會等待狀態中的節點轉移到 `POWER_SAVING` 狀態並變為可用。

為工作配置的所有節點都可用之後，工作就會進入 `RUNNING (R)` 狀態。

依預設，所有工作都會提交至預設佇列 (在中稱為分割區 `Slurm`)。這是由隊列名稱後的後 \* 綴表示。您可以使用 `-p` 工作提交選項來選取佇列。

所有節點都配置了以下功能，可用於工作提交命令：

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

您可以使用以下命令查看特定節點的功能：

```
$ scontrol show nodes nodename
```

在返回中，檢查 `AvailableFeatures` 列表。

考慮叢集的初始狀態，您可以透過執行 `sinfo` 命令來檢視此狀態。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4     idle~ efa-dy-efacompute1-[1-4]
```



```

efa          up    infinite    1    idle efa-st-efacompute1-1
gpu          up    infinite    10   idle~ gpu-dy-gpucompute1-[1-10]
ondemand    up    infinite    20   idle~ ondemand-dy-ondemandcompute1-
[1-10],ondemand-dy-ondemandcompute2-[1-10]
spot*       up    infinite    13   idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*       up    infinite    2    idle spot-st-spotcompute2-[1-2]

```

請注意，這spot是預設佇列。它由\*後綴表示。

將工作提交至預設佇列 (spot) 中的一個靜態節點。

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

將工作提交至EFA佇列中的一個動態節點。

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

將工作提交至ondemand佇列中的八 (8) 個c5.2xlarge節點和兩 (2) 個t2.xlarge節點。

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

將工作提交至gpu佇列中的一個 GPU 節點。

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

使用squeue命令考慮作業的狀態。

```

$ squeue
JOBID PARTITION  NAME  USER  ST      TIME  NODES  NODELIST(REASON)
  12  ondemand  wrap  ubuntu CF      0:36    10  ondemand-dy-ondemandcompute1-
[1-8],ondemand-dy-ondemandcompute2-[1-2]
  13      gpu    wrap  ubuntu CF      0:05     1  gpu-dy-gpucompute1-1
   7      spot  wrap  ubuntu R      2:48     1  spot-st-spotcompute2-1
   8      efa    wrap  ubuntu R      0:39     1  efa-dy-efacompute1-1

```

工作 7 和 8 (在spot和efa佇列中) 已在執行中 (R)。作業 12 和 13 仍在設定 (CF)，可能正在等待執行個體可用。

```
# Nodes states corresponds to state of running jobs
```

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa       up    infinite   3  idle~ efa-dy-efacompute1-[2-4]
efa       up    infinite   1  mix  efa-dy-efacompute1-1
efa       up    infinite   1  idle  efa-st-efacompute1-1
gpu       up    infinite   1  mix~  gpu-dy-gpucompute1-1
gpu       up    infinite   9  idle~ gpu-dy-gpucompute1-[2-10]
ondemand  up    infinite  10  mix#  ondemand-dy-ondemandcompute1-[1-8],ondemand-
dy-ondemandcompute2-[1-2]
ondemand  up    infinite  10  idle~  ondemand-dy-ondemandcompute1-[9-10],ondemand-
dy-ondemandcompute2-[3-10]
spot*     up    infinite  13  idle~  spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*     up    infinite   1  mix  spot-st-spotcompute2-1
spot*     up    infinite   1  idle  spot-st-spotcompute2-2

```

## 節點狀態和功能

在大多數情況下，節點狀態會 AWS ParallelCluster 根據本主題稍早描述的雲端節點生命週期中的特定程序進行完全管理。

但是，AWS ParallelCluster 也會取代或終止運作狀態不良的支援執行個體以DOWN及DRAINED狀態和節點中運作狀態不良的節點。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。分Slurm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，磁碟分割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，支援非作用中分割區之節點的所有執行個體都會終止。不會針對非使用中分割區中的節點啟動新執行個體。

## pcluster update-compute-fleet

- 停止計算叢集-執行下列命令時，所有磁碟分割都會轉換到INACTIVE狀態，並且 AWS ParallelCluster 處理程序會將分割區保持在INACTIVE狀態。

```

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

```

- 啟動計算叢集-執行下列命令時，所有分割區都會最初轉換為UP狀態。但是，AWS ParallelCluster 進程不會將分區保持在一個UP狀態。您需要手動更改分區狀態。幾分鐘後，所有靜態節點都會變為可用。請注意，將分區設置為UP不會啟動任何動態容量。

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

執行update-compute-fleet時，您可以執行pcluster describe-compute-fleet命令並檢查Status。以下列出了可能的狀態：

- STOP\_REQUESTED：停止計算叢集要求會傳送至叢集。
- STOPPING：pcluster程序目前正在停止運算叢集。
- STOPPED：pcluster程序完成停止程序，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：啟動計算叢集要求會傳送至叢集。
- STARTING：pcluster程序目前正在啟動叢集。
- RUNNING：該pcluster過程完成了啟動過程，所有分區都處於UP狀態，靜態節點在幾分鐘後可用。
- PROTECTED：此狀態表示某些分割區具有一致的啟動程序失敗。受影響的分割區是非作用 請調查問題，然後執行update-compute-fleet以重新啟用叢集。

## 手動控制佇列

在某些情況下，您可能想要手動控制叢集中的節點或佇列 (稱為中的分割區Slurm)。您可以使用scontrol指令，透過下列常用程序來管理叢集中的節點。

- 啟動POWER\_SAVING狀態中的動態節點

以 su root 使用者身分執行指令：

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

您也可以提交要求特定節點數目的預留位置sleep 1工作，然後依賴Slurm啟動所需節點數目的電源。

- 之前關閉動態節點的電源 [ScaledownIdleTime](#)

我們建議您使用以下指令將動態節點設定DOWN為 su root 使用者：

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

AWS ParallelCluster 會自動終止並重設已關閉的動態節點。

一般而言，我們不建議您將節點設定為POWER\_DOWN直接使用scontrol update nodename=*nodename* state=power\_down指令。這是因為 AWS ParallelCluster 會自動處理斷電過程。

- 停用佇列 (分割區) 或停止特定分割區中的所有靜態節點

使用以下指令將特定佇列設定INACTIVE為 su root 使用者：

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

這樣做會終止所有支援分割區中節點的執行個體。

- 啟用佇列 (分割區)

使用以下指令將特UP定佇列設定為 su root 使用者：

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

## 縮放行為和調整

以下是一般縮放工作流程的範例：

- 排程器會收到需要兩個節點的工作。
- 排程器會將兩個節點轉換為POWER\_UP狀態，並ResumeProgram使用節點名稱呼叫 (例如queue1-dy-spotcompute1-[1-2])。
- ResumeProgram啟動兩個 Amazon EC2 執行個體，並指派的私有 IP 地址和主機名稱queue1-dy-spotcompute1-[1-2]，等待 ResumeTimeout (預設期間為重設節點前 30 分鐘)。
- 執行個體已設定並加入叢集。工作開始在執行個體上執行。
- 工作完成並停止執行。
- 設定完SuspendTime成後 (設定為 [ScaledownIdletime](#))，排程器會將執行個體設定為狀POWER\_SAVING態。排程器接著會設queue1-dy-spotcompute1-[1-2]定為POWER\_DOWN狀態，並SuspendProgram使用節點名稱呼叫。

- `SuspendProgram`被稱為兩個節點。節點會保持`POWER_DOWN`狀態，例如，保持一個`SuspendTimeout` (預設期間為 120 秒 (2 分鐘))。 `clustermgtd`偵測到節點正在關閉電源後，便會終止支援執行個體。然後，它會轉換`queue1-dy-spotcompute1-[1-2]`到閒置狀態並重置私有 IP 地址和主機名稱，以便為將 `future` 的工作啟動。

如果事情出錯，並且由於某種原因無法啟動特定節點的實例，則會發生以下情況：

- 排程器會收到需要兩個節點的工作。
- 排程器會將兩個雲端成組分解節點轉換為`POWER_UP`狀態，並`ResumeProgram`使用節點名稱呼叫 (例如)。`queue1-dy-spotcompute1-[1-2]`
- `ResumeProgram`僅啟動一 (1) 個 Amazon EC2 執行個體`queue1-dy-spotcompute1-1`，並使用一 (1) 個執行個體進行設定`queue1-dy-spotcompute1-2`，無法啟動。
- `queue1-dy-spotcompute1-1`到達`POWER_UP`後不受影響並上線。
- `queue1-dy-spotcompute1-2`轉換為狀`POWER_DOWN`態，並且工作會自動重新報價，因為Slurm偵測到節點故障。
- `queue1-dy-spotcompute1-2`之後變為可用 `SuspendTimeout` (預設值為 120 秒 (2 分鐘))。在此期間，工作會重新排序，並且可以在另一個節點上開始執行。
- 上述程序會重複執行，直到工作可以在可用的節點上執行而不會發生失敗為止。

有兩個計時參數可以根據需要進行調整：

- **`ResumeTimeout`**(預設值為 30 分鐘)： `ResumeTimeout`控制將節點轉換為停止Slurm狀態之前的等待時間。
  - `ResumeTimeout`如果您的安裝前/後安裝過程需要接近那麼長的時間，擴展可能會很有用。
  - `ResumeTimeout`也是發生問題時，取代或重設節點之前 AWS ParallelCluster 等待的最長時間。如果在啟動或安裝期間發生任何錯誤，計算節點會自動終止。 AWS ParallelCluster 進程在檢測到終止的實例時替換節點。
- **`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`表示節點重設速度更快，並且Slurm可以嘗試更頻繁地啟動執行個體。
  - 較長`SuspendTimeout`意味著失敗的節點重置速度會更慢。同時，Slurm嘗試使用其他節點。如果`SuspendTimeout`超過幾分鐘，請Slurm嘗試在系統中的所有節點之間循環。較長的時間對於大型系統 ( 超過 1,000 個節點 ) 來說`SuspendTimeout`可能有利於減少嘗試頻繁重新排入失敗的作業Slurm時的 stress。

- 請注意，SuspendTimeout不指 AWS ParallelCluster 等待終止節點後備實例的時間。POWER\_DOWN節點的支援執行個體會立即終止。終止程序通常會在幾分鐘內完成。但是，在此期間，節點會保持POWER\_DOWN狀態，無法供排程器使用。

## 架構的日誌

下列清單包含金鑰記錄檔。與 Amazon CloudWatch 日誌一起使用的日誌流名稱具有以下格式 `{hostname}.{instance_id}.{logIdentifier}`，其中 `log Identifier` 遵循日誌名稱。

- 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)

## 常見問題和如何調試：

### 無法啟動、開啟電源或加入叢集的節點

- 動態節點：
  - 檢查記ResumeProgram錄檔，看看ResumeProgram是否已透過節點呼叫。如果沒有，請檢查slurmctld日誌以確定是否Slurm嘗試ResumeProgram使用節點調用。請注意，不正確的權限ResumeProgram可能會導致它無訊息地失敗。
  - 如果呼ResumeProgram叫，請檢查是否已針對節點啟動執行個體。如果執行個體沒有啟動，應該會有明確的錯誤訊息，說明執行個體無法啟動的原因。
  - 如果已啟動執行個體，則在啟動程序期間可能發生一些問題。從記錄檔中尋找對應的私有 IP 位址和執行個體 ID，並在記錄檔中查看特定執行個體的對應啟動程序 CloudWatch 記錄。ResumeProgram
- 靜態節點：
  - 檢查記clustermgtd錄檔，查看是否已針對節點啟動執行個體。如果執行個體未啟動，則執行個體無法啟動的原因應該會有明確的錯誤。
  - 如果已啟動執行個體，則啟動程序會有一些問題。從記錄檔中尋找對應的私有 IP 和執行個體 ID，並在記錄檔中查看特定執行個體的對應啟動程序 CloudWatch 記錄。clustermgtd

## 節點意外取代或終止，以及節點故障

- 節點意外取代/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clustermgtd取代或終止節點，請檢查記clustermgtd錄檔。
  - 如果clustermgtd取代或終止節點，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，節點是DOWN)，請檢查slurmctld記錄檔以取得更多詳細資訊。如果原因與 Amazon EC2 相關，請使用 Amazon CloudWatch 或 Amazon EC2 主控台、CLI 或開發套件等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 Amazon EC2 運作狀態檢查失敗。
  - 如果clustermgtd沒有終止節點，請檢查節點是否終computemgtd止，或者 EC2 是否終止執行個體以回收 Spot 執行個體。
- 節點故障：
  - 在大多數情況下，如果節點故障，工作會自動重新報價。查看slurmctld日誌以查看工作或節點失敗的原因，並從該處評估情況。

## 取代或終止執行個體時失敗，關閉節點電源時失敗

- 一般而言，clustermgtd處理所有預期的執行個體終止動作。查看clustermgtd日誌以查看為什麼無法替換或終止節點。
- 對於動態節點失敗 [ScaledownIdleTime](#)，請查看SuspendProgram日誌以查看slurmctld進程是否以特定節點作為參數進行了調用。Note 實際上SuspendProgram並沒有執行任何特定的操作。相反，它只會在調用時記錄。所有實例終止和NodeAddr重設均由完成clustermgtd。Slurm將節點轉換為IDLE之後SuspendTimeout。

## 其他問題：

- AWS ParallelCluster 不會做出任務分配或擴展決策。它只會嘗試根據指示啟動、終止和維護資源。Slurm

有關工作分配、節點分配和擴展決策的問題，請查看slurmctld日誌中是否有錯誤。

## Slurm叢集保護模式

在啟用受保護模式的情況下執行叢集時，會在啟動運算節點時 AWS ParallelCluster 監視和追蹤計算節點啟動程序失敗。它這樣做是為了檢測這些故障是否持續發生。

如果在佇列 (分割區) 中偵測到下列項目，叢集就會進入受保護狀態：

1. 連續的計算節點啟動程序失敗會持續發生，而不會成功啟動計算節
2. 失敗計數達到預先定義的臨界值。

叢集進入受保護狀態後，AWS ParallelCluster 停用失敗達到或超過預先定義臨界值的佇列。

Slurm在 AWS ParallelCluster 版本 3.0.0 中添加了集群保護模式。

您可以使用受保護模式來減少耗費在計算節點啟動程序失敗循環上的時間和資源。

受保護模式參數

### **protected\_failure\_count**

`protected_failure_count`指定佇列 (分割區) 中啟動叢集保護狀態的連續失敗次數。

預設值`protected_failure_count`為 10，且已啟用受保護模式。

如果大`protected_failure_count`於零，則會啟用受保護模式。

如果小`protected_failure_count`於或等於零，則會停用受保護模式。

您可以透過在位於的`clustermgtd`組態檔案中加入參數來`/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf`變更`protected_failure_count`值`HeadNode`。

您可以隨時更新此參數，而且不需要停止運算叢集即可執行此操作。如果在失敗計數達到之前佇列中啟動成功`protected_failure_count`，失敗計數會重設為零。

檢查叢集狀態為受保護狀態

當叢集處於受保護狀態時，您可以檢查運算叢集狀態和節點狀態。

運算叢集狀態

運算叢集的狀態是PROTECTED在執行受保護狀態的叢集中。

```
$ pcluster describe-compute-fleet --cluster-name <cluster-name> --region <region-id>
{
  "status": "PROTECTED",
```



```
"lastStatusUpdateTime": "2022-04-22T00:31:24.000Z"
}
```

## 節點狀態

若要瞭解哪些佇列 (分割區) 具有啟動受保護狀態的啟動程序失敗，請登入叢集並執行 `sinfo` 命令。啟動程序失敗等於或以上 `protected_failure_count` 的分割區處於 INACTIVE 狀態。沒有啟動程序失敗的磁碟分割處 `protected_failure_count` 於 UP 狀態，並如預期般運作。

PROTECTED 狀態不會影響正在執行的工作。如果工作在具有啟動程序失敗等於或以上的分割區上執行 `protected_failure_count`，則在執行中的工作完成 INACTIVE 後，該分割區會設定為。

請考慮下列範例中顯示的節點狀態。

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   inact infinite 10  down% queue1-dy-c5xlarge-[1-10]
queue1*   inact infinite 3490 idle~ queue1-dy-c5xlarge-[11-3500]
queue2    up   infinite 10  idle~ queue2-dy-c5xlarge-[1-10]
```

分割區 `queue1` 是 INACTIVE 因為偵測到 10 個連續的計算節點啟動程序失敗。

節點後面的執行個體 `queue1-dy-c5xlarge-[1-10]` 已啟動但無法加入叢集，因為狀態不良。

叢集處於受保護狀態。

分割區 `queue2` 不受中的啟動程序失敗影響 `queue1`。它處於 UP 狀態，仍然可以運行作業。

## 如何停用受保護狀態

解決啟動程序錯誤之後，您可以執行下列命令，將叢集移出受保護狀態。

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

## 啟動受保護狀態的引導失敗

激活受保護狀態的引導錯誤細分為以下三種類型。要識別類型和問題，您可以檢查是否 AWS ParallelCluster 生成日誌。如果已產生記錄檔，您可以檢查它們以取得錯誤詳細資訊。如需詳細資訊，請參閱 [擷取和保留記錄](#)。

## 1. 引導程序錯誤，導致實例自我終止。

執行個體會在啟動程序的早期失敗，例如因為 [SlurmQueues\CustomActions\OnNodeStart\OnNodeConfigured](#) 指令碼中的錯誤而自行終止的執行個體。

對於動態節點，請尋找類似下列內容的錯誤：

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

對於靜態節點，請在 `clustermgtd log (/var/log/parallelcluster/clustermgtd)` 中查看類似以下內容的錯誤：

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

## 2. 節點 `resume_timeout` 或 `node_replacement_timeout` 過期。

執行個體無法在 `resume_timeout` (針對動態節點) 或 `node_replacement_timeout` (針對靜態節點) 內加入叢集。它不會在超時之前自行終止。例如，未正確設定叢集的網路，而且在逾時到期 Slurm 之後，節點會設定為 DOWN 狀態。

對於動態節點，請尋找類似下列內容的錯誤：

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

對於靜態節點，請在 `clustermgtd log (/var/log/parallelcluster/clustermgtd)` 中查看類似以下內容的錯誤：

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

## 3. 節點運作狀態檢查失敗。

節點後面的執行個體無法進行 Amazon EC2 運作狀態檢查或排程的事件運作狀態檢查，並將節點視為啟動程序故障節點。在此情況下，執行個體會因無法控制的 AWS ParallelCluster 原因而終止。

在 `clustermgtd log (/var/log/parallelcluster/clustermgtd)` 中查看類似下列內容的錯誤：

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

## 4. 計算節點 Slurm 註冊失敗。

向Slurm控制slurmd常駐程式 (slurmctld) 註冊常駐程式失敗，並導致計算節點狀態變更為INVALID\_REG狀態。不正確設定的Slurm運算節點可能會造成此錯誤，例如設定了[CustomSlurmSettings](#)運算節點規格錯誤的計算節點。

查看 head 節點上的slurmctld記錄檔 (/var/log/slurmctld.log)，或查看失敗計算節點的slurmd記錄檔 (/var/log/slurmd.log)，找出類似下列內容的錯誤：

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

## 如何調試受保護模式

如果您的叢集處於受保護狀態，並且從有問題的計算節點 AWS ParallelCluster 產生clustermgtdcloud-init-output記錄檔HeadNode和記錄檔，則您可以檢查記錄檔以取得錯誤詳細資訊。如需如何擷取記錄檔的詳細資訊，請參閱[擷取和保留記錄](#)。

### clustermgtd標頭節點上的 log ( /var/log/parallelcluster/clustermgtd )

記錄訊息會顯示哪些分割區有啟動程序失敗，以及對應的啟動程序失敗

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - INFO - Partitions bootstrap failure count: {'queue1': 2}, cluster will be set into protected mode if protected failure count reach threshold.
```

在clustermgtd記錄檔中，搜尋Found the following bootstrap failure nodes以尋找啟動失敗的節點。

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - WARNING - Found the following bootstrap failure nodes: (x2) ['queue1-st-c5large-1(192.168.110.155)', 'broken-st-c5large-2(192.168.65.215)']
```

在記clustermgtd錄檔中，搜尋Node bootstrap error以尋找失敗的原因。

```
[slurm_plugin.clustermgtd:_is_node_bootstrap_failure] - WARNING - Node bootstrap error: Node broken-st-c5large-2(192.168.65.215) is currently in replacement and no backing instance
```

### cloud-init-output運算節點上的 log (/var/log/cloud-init-output.log)

在記錄檔中取得啟動程序失敗節點私人 IP 位址之後，您可以登入計算節點或遵循中的指示擷取記錄檔，[擷取和保留記錄](#)來尋找對應的計算節點記錄。clustermgtd在大多數情況下，有問題節點的/var/log/cloud-init-output記錄檔會顯示造成計算節點啟動程序失敗的步驟。

## Slurm叢集快速容量不足容錯移轉

從 AWS ParallelCluster 版本 3.2.0 開始，叢集會在預設啟用快速容量不足的容錯移轉模式下執行。如此可將偵測到 Amazon EC2 容量不足錯誤時，重試將任務排入佇列所花費的時間降至最低。當您使用多種執行個體類型設定叢集時，這特別有效。

Amazon EC2 檢測到容量不足故障：

- InsufficientInstanceCapacity
- InsufficientHostCapacity
- InsufficientReservedInstanceCapacity
- MaxSpotInstanceCountExceeded
- SpotMaxPriceTooLow：如果 Spot 請求價格低於所需的最低 Spot 請求履行價格，則啟用。
- Unsupported：使用特定不支援的執行個體類型來啟用 AWS 區域。

在快速容量不足的容錯移轉模式下，如果在工作指派給 [SlurmQueues](#)/時偵測到容量不足錯誤 [compute resource](#)，請 AWS ParallelCluster 執行下列動作：

1. 它將計算資源設置為 disabled ( DOWN ) 狀態的預定時間段。
2. 它會用POWER\_DOWN\_FORCE來取消失敗節點工作的計算資源，並暫停失敗的節點。它會將失敗節點設定為IDLE和POWER\_DOWN (!)狀態，然後設定為POWERING\_DOWN (%)。
3. 它會將工作重新排序至另一個計算資源。

停用之運算資源的靜態和已啟動節點不會受到影響。工作可以在這些節點上完成。

此週期會重複，直到工作成功指派給一或多個計算資源節點為止。如需節點狀態的相關資訊，請參閱[Slurm多隊列模式指南](#)。

如果找不到可執行工作的計算資源，則工作會設定為PENDING狀態，直到經過預先定義的時間為止。在此情況下，您可以依照下一節所述修改預先定義的時間週期。

容量逾時參數不足

### **insufficient\_capacity\_timeout**

`insufficient_capacity_timeout` 指定偵測到容量不足錯誤時，計算資源保持在 disabled (down) 狀態的時間 (以秒為單位)。

依預設，`insufficient_capacity_timeout` 為啟用狀態。

預設值 `insufficient_capacity_timeout` 為 600 秒 (10 分鐘)。

如果該 `insufficient_capacity_timeout` 值小於或等於零，將停用快速不足容量容錯移轉模式。

您可以在 `insufficient_capacity_timeout` 位於的 `clustermgtd` 組態檔案中加入參數來變更值 `HeadNode`。 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf`

您可以隨時更新參數，而無需停止運算叢集。

例如：

- `insufficient_capacity_timeout=600`:

如果偵測到容量不足錯誤，則計算資源會設定為 disabled (DOWN)。10 分鐘後，其失敗的節點會設定為 `idle~ (POWER_SAVING)` 狀態。

- `insufficient_capacity_timeout=60`:

如果偵測到容量不足錯誤，則計算資源位於 disabled (DOWN) 中。1 分鐘後，其失敗的節點會設定為 `idle~` 狀態。

- `insufficient_capacity_timeout=0`:

快速容量不足的容錯移轉模式已停用。未停用計算資源。

#### Note

節點故障且容量不足錯誤與叢集管理協助程式偵測到節點故障的時間之間，最多可能會有一分鐘的延遲時間。這是因為叢集管理常駐程式會檢查節點容量不足故障，並以一分鐘的間隔將計算資源設定為 `down` 狀態。

快速容量不足的容錯移轉模式狀態

當叢集處於快速容量不足的容錯移轉模式時，您可以檢查其狀態和節點狀態。

## 節點狀態

當工作提交至計算資源動態節點並偵測到容量不足錯誤時，節點會以原因置於down#狀態。

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

然後關閉電源的節點 ( idle~狀態中的節點 ) 被設置為down~有原因。

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

工作會重新計算至佇列中的其他計算資源。

不受快速容量不足容量容錯移轉模式影響的計算資源靜態節點和節點。UP

請考慮下列範例中顯示的節點狀態。

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

我們提交一個作業到 queue1 需要一個節點。

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    1   down# queue1-dy-c-1-1
queue1*   up    infinite    15  idle~ queue1-dy-c-2-[1-15]
queue1*   up    infinite    14  down~ queue1-dy-c-1-[2-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

節點queue1-dy-c-1-1已啟動以執行工作。不過，執行個體無法啟動，因為容量不足錯誤。節點queue1-dy-c-1-1設定為down。計算資源 (queue2-dy-c-1) 中已關閉電源的動態節點設定為down。

您可以使用檢查節點原因scontrol show nodes。

```
$ scontrol show nodes queue1-dy-c-1-1
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

```
Reason=(Code:InsufficientInstanceCapacity)Failure when resuming nodes
[root@2022-03-10T22:17:50]

$ scontrol show nodes queue1-dy-c-1-2
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Temporarily disabling node due to
insufficient capacity [root@2022-03-10T22:17:50]
```

工作會排入佇列運算資源中的其他執行個體類型。

`insufficient_capacity_timeout`經過之後，計算資源中的節點會重設為狀`idle~`態。

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

計算資源中的`insufficient_capacity_timeout`經歷和節點重設為`idle~`狀態後，Slurm排程器會將節點的優先順序降低。排程器會繼續從其他佇列計算資源中選取具有較高權數的節點，除非發生下列其中一種情況：

- 工作的提交需求符合復原的計算資源。
- 沒有其他可用的運算資源，因為它們正在運作。
- `slurmctld`重新啟動。
- AWS ParallelCluster 運算叢集會停止並開始關閉電源並開啟所有節點的電源。

## 相關記錄

與容量不足錯誤和快速容量不足容量容錯移轉模式相關的日誌可以在頭Slurm節點的日`resume`誌和日`clustermgtd`誌中找到。

## Slurm `resume (/var/log/parallelcluster/slurm_resume.log)`

當節點因為容量不足而無法啟動時出現錯誤訊息。

```
[slurm_plugin.instance_manager:_launch_ec2_instances] - ERROR - Failed RunInstances
request: dcd0c252-90d4-44a7-9c79-ef740f7ecd87
```

```
[slurm_plugin.instance_manager:add_instances_for_nodes] - ERROR - Encountered exception when launching instances for nodes (x1) ['queue1-dy-c-1-1']: An error occurred (InsufficientInstanceCapacity) when calling the RunInstances operation (reached max retries: 1): We currently do not have sufficient p4d.24xlarge capacity in the Availability Zone you requested (us-west-2b). Our system will be working on provisioning additional capacity. You can currently get p4d.24xlarge capacity by not specifying an Availability Zone in your request or choosing us-west-2a, us-west-2c.
```

## Slurm `clustermgtd` (`/var/log/parallelcluster/clustermgtd`)

佇列 1 中的計算資源 c-1 已停用，因為容量不足。

```
[slurm_plugin.clustermgtd:_reset_timeout_expired_compute_resources] - INFO - The following compute resources are in down state due to insufficient capacity: {'queue1': {'c-1': ComputeResourceFailureEvent(timestamp=datetime.datetime(2022, 4, 14, 23, 0, 4, 769380, tzinfo=datetime.timezone.utc), error_code='InsufficientInstanceCapacity')}}}, compute resources are reset after insufficient capacity timeout (600 seconds) expired
```

容量不足逾時到期後，會重設計算資源，計算資源中的節點會設定為idle~。

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the following compute resources because insufficient capacity timeout expired: {'queue1': ['c-1']}
```

## Slurm 記憶體式排程

從版本 3.2.0 開始，AWS ParallelCluster 支援使用

[SlurmSettings/EnableMemoryBasedScheduling](#) 叢集配置參數的Slurm記憶體型排程。

### Note

從 AWS ParallelCluster 版本 3.7.0 開始，如果您在執行個體中設定多個[執行](#)個體類型，則EnableMemoryBasedScheduling可以啟用此功能。

適用於 AWS ParallelCluster 版本 3.2.0 至 3.6 版本。

**x**，EnableMemoryBasedScheduling如果您在執行個體中設定多個[執行](#)個體類型，則無法啟用。



**⚠ Warning**

當您在Slurm佇列運算資源中指定多個執行個體類型且EnableMemoryBasedScheduling已啟用時，該RealMemory值是可供所有執行個體類型使用的最小記憶體數量。如果您指定的記憶體容量非常不同的執行個體類型，這可能會導致大量未使用的記憶體。

使用時EnableMemoryBasedScheduling: true，Slurm排程器會追蹤每個工作在每個節點上所需的記憶體數量。然後，Slurm排程器會使用此資訊，在同一個計算節點上排程多個工作。節點上作業所需的記憶體總容量不能大於可用節點記憶體。排程器可防止工作使用比提交工作時要求的記憶體更多。

使用時EnableMemoryBasedScheduling: false，工作可能會爭取共用節點上的記憶體，並造成工作失敗和out-of-memory事件。

**⚠ Warning**

Slurm對其標籤使用 2 個符號的冪，例如 MB 或 GB。將這些標籤分別讀取為 MiB 和 GiB。

## Slurm組態和記憶體型排程

使用EnableMemoryBasedScheduling: true，Slurm設定下列Slurm組態參數：

- [SelectTypeParameters=CR\\_CPU\\_Memory](#) (位於《slurm.conf》)。此選項會將節點記憶體設定為中的可消耗資源。Slurm
- [ConstrainRAMSpace=yes](#)在中Slurmcgroup.conf。使用此選項時，工作對記憶體的存取會限制在提交作業時要求的記憶體數量。

**📘 Note**

設定這兩個選項時，其他數個Slurm組態參數可能會影響Slurm排程器和資源管理員的行為。如需詳細資訊，請參閱 [Slurm 文件](#)。

## Slurm排程器和記憶體型排程

**EnableMemoryBasedScheduling: false**(預設值)

默認情況下EnableMemoryBasedScheduling，設置為假。如果為 false，則Slurm不會在其排程演算法中包含記憶體做為資源，也不會追蹤作業使用的記憶體。使用者可以指定--mem MEM\_PER\_NODE選項，以設定工作所需的每個節點的最小記憶體容量。這會強制排程器在排程工作MEM\_PER\_NODE時選擇RealMemory值至少為的節點。

例如，假設使用者使用提交兩個工作。--mem=5GB如果要求的資源 (例如 CPU 或 GPU) 可供使用，則作業可以在具有 8 GiB 記憶體的節點上同時執行。這兩個作業不會排程在低於 5 GiB 的RealMemory運算節點上。

#### Warning

停用記憶體型排程時，Slurm不會追蹤工作使用的記憶體量。在相同節點上執行工作可能會爭取記憶體資源，並導致其他工作失敗。

停用記憶體型排程時，建議使用者不要指定--mem-per-cpu或--mem-per-gpu選項。這些選項可能會導致行為與說明[Slurm文件](#)中描述的不同。

### EnableMemoryBasedScheduling: true

設定EnableMemoryBasedScheduling為 true 時，會Slurm追蹤每個工作的記憶體使用量，並防止工作使用超過--mem提交選項要求的記憶體。

在前面的範例中，使用者使用提交兩個工作。--mem=5GB作業無法在具有 8 GiB 記憶體的節點上同時執行。這是因為所需的記憶體總量大於節點上可用的記憶體。

啟用記憶體型排程後，--mem-per-cpu並與Slurm文件中所述的--mem-per-gpu行為一致。例如，使用提交工作--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB。在此情況下，Slurm會為每個節點指派工作總共 4 GiB。

#### Warning

啟用記憶體式排程時，建議使用者在提交工作時加入--mem規格。使用隨附的預設Slurm組態AWS ParallelCluster，如果沒有包含記憶體選項(--mem--mem-per-cpu、或--mem-per-gpu)，則Slurm會將已配置節點的整個記憶體指派給作業，即使它只要求部分其他資源 (例如 CPU 或 GPU) 也一樣。這有效地防止節點共用，直到工作完成，因為沒有記憶體可供其他工作使用。發生這種情況是因Slurm為在工作提交DefMemPerNode時未提供記憶體規格時，將工作每個節點的記憶體設定為。此參數的預設值為 0，並指定對節點記憶體的無限制存取。

如果同一佇列中有多種具有不同記憶體數量的運算資源，則不含記憶體選項的工作可能會在不同節點上指派不同數量的記憶體。這取決於排程器可供工作使用的節點。使用者可以

在Slurm組態檔案中的叢集DefMemPerNode或磁碟分割層級為選項 (例如或) 定義自訂值 [DefMemPerCPU](#)，以防止此行為發生。

## Slurm RealMemory 和 AWS ParallelCluster SchedulableMemory

使用隨附的Slurm組態 AWS ParallelCluster，將Slurm解譯[RealMemory](#)為每個節點可供工作使用的記憶體容量。從版本 3.2.0 開始，依預設，會 AWS ParallelCluster 設定RealMemory為 [Amazon EC2 執行個體類型](#)中列出的 95% 記憶體，並由 Amazon EC2 API [DescribeInstanceTypes](#)傳回。

停用記憶體型排程時，排程Slurm器會在使用者提交具有--mem指定的工作時使用RealMemory來篩選節點。

啟用記憶體型排程時，排程Slurm器會解譯RealMemory為運算節點上執行的工作可用的最大記憶體容量。

預設設定可能不是所有執行個體類型的最佳選擇：

- 此設定可能會高於節點實際存取的記憶體數量。當計算節點是小型執行個體類型時，就會發生這種情況。
- 此設定可能低於節點實際可存取的記憶體數量。當計算節點是大型執行個體類型時，就可能會發生這種情況，並且可能導致大量未使用的記憶體。

您可以使用 [SlurmQueuesComputeResources//SchedulableMemory](#)微調運算節點所RealMemory AWS ParallelCluster 配置的值。若要覆寫預設值，請為叢集配置SchedulableMemory專門定義自訂值。

若要檢查計算節點的實際可用記憶體，請在節點上執行/opt/slurm/sbin/slurmd -C命令。此命令返回節點的硬件配置，包括[RealMemory](#)值。如需詳細資訊，請參閱 [slurmd -C](#)。

請確定計算節點的作業系統處理序有足夠的記憶體。若要執行此操作，請將值設定為低於slurmd -C指令傳回的SchedulableMemory值，以限制工作可用的記憶體。RealMemory

## 使用 Slurm 進行多個實例類型分配

從 3.3.0 AWS ParallelCluster 版開始，您可以將叢集設定為從計算資源的已定義執行個體類型集進行配置。配置可以根據 Amazon EC2 叢集低成本或最佳容量策略進行配置。

這組已定義的執行個體類型必須全部具有相同數量的 vCPUs，或者如果停用多執行緒，則必須具有相同數量的核心。此外，這組執行個體類型必須具有相同製造商的加速器數目相同。如

果將 `Efa`/設定 `Enabled` 為 `true`，則執行個體必須支援 EFA。如需詳細資訊和需求，請參閱 [SchedulingSlurmQueues//AllocationStrategy](#) 和 [ComputeResources/Instances](#)。

您可以 `capacity-optimized` 根據您的 `CapacityType` 組態設定 `AllocationStrategy` 為 `lowest-price` 或。

在中 [Instances](#)，您可以設定一組執行個體類型。

### Note

從 AWS ParallelCluster 版本 3.7.0 開始，如果您在執行個體中設定多個 `執行個體類型`，則 `EnableMemoryBasedScheduling` 可以啟用此功能。

適用於 AWS ParallelCluster 版本 3.2.0 至 3.6 版本。

**x**，`EnableMemoryBasedScheduling` 如果您在執行個體中設定多個 `執行個體類型`，則無法啟用。

下列範例說明如何查詢 vCPUs、EFA 支援和架構的執行個體類型。

`InstanceTypes` 使用 96 個 vCPUs 和 `x86_64` 架構進行查詢。

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-vcpus,Values=96" "Name=processor-info.supported-
  architecture,Values=x86_64" \
  --query "sort_by(InstanceTypes[*]).
  {InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
  &InstanceType)" \
  --output table
```

`InstanceTypes` 使用 64 個核心、EFA 支援和 `arm64` 架構進行查詢。

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-cores,Values=64" "Name=processor-
  info.supported-architecture,Values=arm64" "Name=network-info.efa-
  supported,Values=true" --query "sort_by(InstanceTypes[*]).
  {InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
  &InstanceType)" \
  --output table
```

下一個範例叢集配置程式碼片段顯示如何使用這些 `InstanceType` 和 `AllocationStrategy` 屬性。

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue-1
      CapacityType: ONDEMAND
      AllocationStrategy: lowest-price
      ...
      ComputeResources:
        - Name: computeresource1
          Instances:
            - InstanceType: r6g.2xlarge
            - InstanceType: m6g.2xlarge
            - InstanceType: c6g.2xlarge
          MinCount: 0
          MaxCount: 500
        - Name: computeresource2
          Instances:
            - InstanceType: m6g.12xlarge
            - InstanceType: x2gd.12xlarge
          MinCount: 0
          MaxCount: 500
  ...
...
```

## 動態節點的叢集擴展

ParallelCluster 支持通過使用Slurm的省電插件動態擴展集群Slurm的方法。如需詳細資訊，請參閱Slurm文件中的[雲端排程指南和省Slurm電指南](#)。

從 3.8.0 ParallelCluster 版開始，ParallelCluster 使用 Job 層級履歷或工作層級調整作為預設動態節點配置策略來擴展叢集：ParallelCluster 根據每個作業的需求、配置給作業的節點數目以及需要恢復的節點來擴展叢集。ParallelCluster 從「文件」環境變量中獲取此信息。

動態節點的擴展程序為兩個步驟，包括啟動 EC2 執行個體，以及將已啟動的 Amazon EC2 執行個體指派到Slurm節點。這兩個步驟中的每一個都可以使用all-or-nothing或盡力邏輯來完成。

若要啟動亞 Amazon EC2 執行個體：

- all-or-nothing調用啟動 Amazon EC2 API 的最小目標等於總目標容量
- 盡最大努力呼叫啟動 Amazon EC2 API 的最小目標等於 1，且目標容量總計等於請求的容量

對於將 Amazon EC2 執行個體指派到 Slurm 節點：

- all-or-nothing 只有在可以將 Amazon EC2 執行個體指派給每個請求的 Slurm 節點時，才會將 Amazon EC2 執行個體指派給節點
- 盡最大努力將 Amazon EC2 執行個體指派給 Slurm 節點，即使 Amazon EC2 執行個體容量未涵蓋所有請求的節點

上述策略的可能組合轉化為 ParallelCluster 啟動策略。

## Example

<caption>The available ParallelCluster 啟動策略 that can be set into the [ScalingStrategy](#) cluster configuration to be used with 工作層級縮放 are:</caption>

all-or-nothing 縮放：

此策略涉及為每個任務 AWS ParallelCluster 啟動 Amazon EC2 啟動執行個體 API 呼叫，這需要成功啟動請求的運算節點所需的所有執行個體。這樣可確保叢集只有在每個作業所需的容量可用時才會擴展，避免在擴展程序結束時留下閒置的執行個體。

該策略使用 all-or-nothing 邏輯為每個任務啟動 Amazon EC2 執行個體，以及將 Amazon EC2 執行個體指派到 Slurm 節點的 all-or-nothing 邏輯。

策略群組會以批次方式啟動請求，每個請求的計算資源各一個，每個節點最多 500 個。對於跨越多個計算資源或超過 500 個節點的請求，ParallelCluster 依序處理多個批次。

任何單一資源的批次失敗會導致所有相關聯的未使用產能終止，確保在擴展程序結束時不會留下閒置的執行個體。

## 限制

- 縮放所花費的時間與每次執行 Slurm 恢復程式所提交的工作數量成正比。
- 擴展操作受 RunInstances 資源帳號限制的限制，依預設設定為 1000 個執行個體。此限制是根據 EC2 API 節流政策進行的，如需詳細資訊，請參閱 [Amazon EC2 API 節流文件](#) AWS
- 當您在跨多個可用區域的佇列中，在具有單一執行個體類型的運算資源中提交任務時，只有在單一可用區域中提供所有容量時，all-or-nothing EC2 啟動 API 呼叫才會成功。
- 當您在具有單一可用區域的佇列中，在具有多個執行個體類型的運算資源中提交任務時，只有在單一執行個體類型可提供所有容量時，all-or-nothing Amazon EC2 啟動 API 呼叫才會成功。
- 當您在具有多個執行個體類型的運算資源中提交任務時，在跨多個可用區域的佇列中，不支援 all-or-nothing Amazon EC2 啟動 API 呼叫，而是以最佳方式 ParallelCluster 進行擴展。

## greedy-all-or-nothing縮放：

該 all-or-nothing 策略的這種變體仍可確保叢集只有在每個任務所需的容量可用時才擴展，避免在擴展程序結束時發生閒置執行個體，但是它涉及 ParallelCluster 啟動旨在最小目標容量為 1 的 Amazon EC2 啟動執行個體 API 呼叫，嘗試將啟動到所請求容量的節點數量最大化。該策略使用盡最大努力的邏輯來啟動所有任務的 EC2 執行個體，加上將 Amazon EC2 執行個體指派到每個任務的 Slurm 節點的 all-or-nothing 邏輯。

策略群組會以批次方式啟動請求，每個請求的計算資源各一個，每個節點最多 500 個。對於跨越多個計算資源或超過 500 個節點的請求，ParallelCluster 會依序處理多個批次。

它可確保在擴展程序結束時不會留下任何閒置的執行個體，方法是在擴展程序期間最大化輸送量，而代價是暫時過度擴展的代價。

### 限制

- 暫時過度擴展是可能的，導致執行個體在擴展完成之前轉換為執行中狀態的額外成本。
- 與 all-or-nothing 策略中 AWS 的執行個體限制相同，需視乎 RunInstances 資源帳號限制而定。

## 盡最大努力擴展：

此策略會呼叫 Amazon EC2 啟動執行個體 API 呼叫，其目標是以最小容量為 1 的目標來達成請求的總容量，但在執行擴展程序後離開閒置執行個體 (如果並非所有請求的容量都可用)。該策略使用盡最大努力的邏輯來啟動所有任務的 Amazon EC2 執行個體，加上將 Amazon EC2 執行個體指派給每個任務的 Slurm 節點的最大努力邏輯。

策略群組會以批次方式啟動請求，每個請求的計算資源各一個，每個節點最多 500 個。對於跨越多個計算資源或超過 500 個節點的請求，ParallelCluster 依序處理多個批次。

此策略允許在多個擴展程序執行上擴展到遠遠超過預設 1000 個執行個體限制，代價是在不同的擴展流程中使用閒置執行個體。

### 限制

- 在擴展程序結束時，可能會有閒置執行個體，適用於無法配置作業要求的所有節點的情況。

以下範例顯示使用不同 ParallelCluster 啟動策略來縮放動態節點的行為。假設您已經提交了兩個任務，每個請求 20 個節點，總共 40 個相同類型的節點，但只有 30 個 Amazon EC2 執行個體可用於涵蓋 EC2 上請求的容量。

### all-or-nothing縮放：

- 對於第一個任務，會呼叫 all-or-nothingAmazon EC2 啟動執行個體 API，請求 20 個執行個體。成功的呼叫會導致啟動 20 個執行個體
- all-or-nothing 成功將 20 個啟動的執行個體指派給第一個工作的Slurm節點
- 另一個 all-or-nothingAmazon EC2 啟動執行個體 API 會呼叫，針對第二個任務請求 20 個執行個體。呼叫不成功，因為只有另外 10 個執行個體的容量。目前沒有啟動任何執行個體

### greedy-all-or-nothing縮放：

- 稱為最大努力的 Amazon EC2 啟動執行個體 API，請求 40 個執行個體，這是所有任務要求的總容量。這導致啟動了 30 個實例
- 將 20 個已啟動的執行個體all-or-nothing指派給第一個工作的Slurm節點成功
- 系統會嘗試將剩餘啟動的執行個體all-or-nothing指派給第二個工作的Slurm節點，但由於工作要求的總共 20 個中只有 10 個可用的執行個體，因此指派不成功
- 終止 10 個未指派啟動的執行個體

### 盡最大努力擴展：

- 稱為最大努力的 Amazon EC2 啟動執行個體 API，請求 40 個執行個體，這是所有任務要求的總容量。這會導致啟動 30 個執行個體。
- 將 20 個已啟動執行個體的最大努力指派給第一個工作的Slurm節點，是成功的。
- 即使要求的總容量為 20，將剩餘 10 個已啟動的執行個體指派給第二個工作的Slurm節點，也會成功完成另一個已啟動的執行個體。但由於該任務是請求 20 個節點，並且可以僅將 Amazon EC2 執行個體指派給其中 10 個節點，因此任務無法啟動且執行個體處於閒置狀態，直到找到足夠的容量以在稍後調用擴展程序時啟動缺少的 10 個執行個體，否則排程器會在其他已經執行的運算節點上排定工作。

### Slurm3.7.x 版中的動態節點分配策略

#### ParallelCluster 使用 2 種類型的動態節點分配策略來擴展叢集：

- 根據可用的請求節點資訊配置：
  - 所有節點恢復或節點列表縮放：



ParallelCluster ResumeProgram運行時Slurm僅根據請求Slurm的節點列表名稱擴展叢集。它僅通過節點名稱將計算資源分配給節點。節點名稱清單可以跨越多個工作。

- 工作層級履歷或工作層級調整：

ParallelCluster 根據每個工作的需求、目前配置給作業的節點數目，以及需要恢復哪些節點來擴展叢集。ParallelCluster 從SLURM\_RESUME\_FILE環境變數取得此資訊。

- 使用 Amazon EC2 啟動策略進行分配：

- 最大努力擴展：

ParallelCluster 透過使用最小目標容量等於 1 的 Amazon EC2 啟動執行個體 API 呼叫來擴展叢集，以啟動支援所請求節點所需的部分但不一定是所有執行個體。

- 一個ll-or-nothing縮放比例：

ParallelCluster 使用 Amazon EC2 啟動執行個體 API 呼叫擴展叢集，該呼叫只有在支援所請求節點所需的所有執行個體都啟動時才會成功。在這種情況下，它會呼叫 Amazon EC2 啟動執行個體 API，其最小目標容量等於請求的總容量。

預設情況下，ParallelCluster 使用節點清單擴展以及最大功率的 Amazon EC2 啟動策略來啟動部分但不一定是支援所請求節點所需的所有執行個體。它會嘗試佈建盡可能多的容量，以滿足提交的工作負載。

從 ParallelCluster 版本 3.7.0 開始，針對以獨佔模式提交的任務，ParallelCluster 使用 all-or-nothingEC2 啟動策略的工作層級擴展。當您以獨佔模式提交工作時，該工作會對其配置的節點具有獨佔存取權。[如需詳細資訊，請參閱Slurm文件中的 EXCLUSIVE。](#)

若要以獨佔模式提交工作：

- 將Slurm工作送出至叢集時，傳送獨佔旗標。例如 `sbatch ... --exclusive`。

或

- 將工作提交至已[JobExclusiveAllocation](#)設定為的叢集佇列true。

以獨佔模式提交工作時：

- ParallelCluster 目前批次會啟動要求，以包含最多 500 個節點。如果工作要求超過 500 個節點，ParallelCluster 則會針對每組 500 個節點提all-or-nothing出啟動要求，並針對其餘節點提出額外的啟動要求。

- 如果節點配置位於單一計算資源中，ParallelCluster 則會針對每組 500 個節點提出 all-or-nothing 啟動請求，並針對其餘節點提出額外的啟動請求。如果啟動要求失敗，則 ParallelCluster 會終止所有啟動要求所建立的未使用容量。
- 如果節點配置跨越多個計算資源，則 ParallelCluster 需要針對每個計算資源提出 all-or-nothing 啟動請求。這些請求也會批次處理。如果其中一個計算資源的啟動要求失敗，則 ParallelCluster 會終止所有計算資源啟動要求所建立的未使用容量。

具有已知限制的 all-or-nothing 啟動策略的工作層級擴展：

- 當您在跨多個可用區域的佇列中，在具有單一執行個體類型的運算資源中提交任務時，只有在單一可用區域中提供所有容量時，all-or-nothing EC2 啟動 API 呼叫才會成功。
- 當您在具有單一可用區域的佇列中，在具有多個執行個體類型的運算資源中提交任務時，只有在單一執行個體類型可提供所有容量時，all-or-nothing Amazon EC2 啟動 API 呼叫才會成功。
- 當您在具有多個執行個體類型的運算資源中提交任務時，在跨多個可用區域的佇列中，不支援 all-or-nothing Amazon EC2 啟動 API 呼叫，而是以最佳方式 ParallelCluster 進行擴展。

Slurm 3.6.x 版及之前版本中的動態節點配置策略

AWS ParallelCluster 僅使用一種類型的動態節點配置策略來擴展叢集：

- 根據可用的請求節點資訊配置：
  - 所有節點恢復或節點列表擴展：ParallelCluster 僅在 Slurm 運行時 Slurm 根據請求的節點列表名稱擴展叢集。ResumeProgram 它僅通過節點名稱將計算資源分配給節點。節點名稱清單可以跨越多個工作。
- 使用 Amazon EC2 啟動策略進行分配：
  - 盡力擴展：ParallelCluster 透過使用最小目標容量等於 1 的 Amazon EC2 啟動執行個體 API 呼叫來擴展叢集，以啟動支援所請求節點所需的部分但不一定是所有執行個體。

ParallelCluster 使用節點清單擴展，並以最大努力的 Amazon EC2 啟動策略來啟動部分但不一定是支援請求節點所需的所有執行個體。它會嘗試佈建盡可能多的容量，以滿足提交的工作負載。

限制

- 在擴展程序結束時，可能會有閒置執行個體，適用於無法配置作業要求的所有節點的情況。

## Slurm會計與 AWS ParallelCluster

從版本 3.3.0 開始，AWS ParallelCluster 支援使用叢集配置參數 [SlurmSettings/資料庫](#) 進行Slurm帳戶。

從版本 3.10.0 開始，AWS ParallelCluster 支援使用具有叢集設定參數/的外部 Slurmdbd 進行Slurm計算。[SlurmSettingsExternalSlurmdbd](#)如果多個叢集共用相同的資料庫，建議使用外部 Slurmdbd。

透過Slurm會計，您可以整合外部會計資料庫以執行下列作業：

- 管理叢集使用者或使用者群組和其他實體。透過此功能，您可以使用更進階Slurm的功能，例如資源限制強制執行、公平共用和 QoS。
- 收集並儲存作業資料，例如執行工作的使用者、工作的持續時間及其使用的資源。您可以使用該sacct實用程序查看保存的數據。

### Note

AWS ParallelCluster 支持支持的 [MySQL 數據庫服務器](#)的SlurmSlurm計算。

Slurmdbd在 AWS ParallelCluster v3.10.0 及更高版本中使用外部Slurm會計工作

在設定Slurm帳戶之前，您必須擁有連線至現有外部Slurmdbd資料庫伺服器的現有外部資料庫伺服器。

若要設定此項目，請定義下列項目：

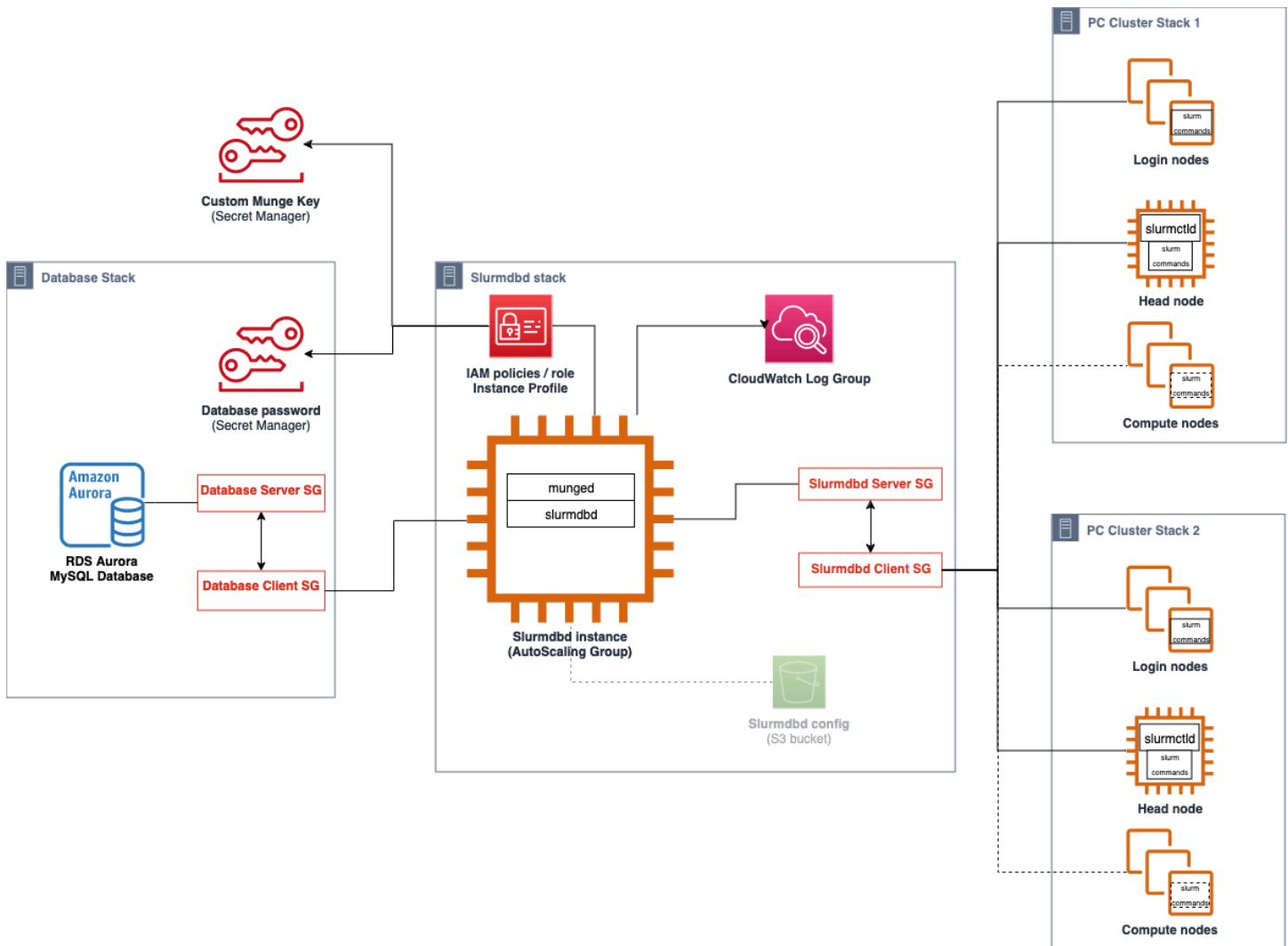
- [ExternalSlurmdbd/Host](#) 中外部Slurmdbd伺服器的位址。伺服器必須存在且可從頭節點存取。
- 在[MungeKeySecretArn](#)中與外部Slurmdbd伺服器通訊的 munge 金鑰。

若要逐步執行自學課程，請參閱[使用外部Slurmdbd帳戶建立叢集](#)。

### Note

您負責管理Slurm資料庫會計實體。

AWS ParallelCluster 外部SlurmDB支援功能的架構可讓多個叢集共用相同SlurmDB和相同的資料庫。



### ⚠ Warning

與外部 AWS ParallelCluster 之間的流SlurmDB量未加密。建議您在受信任的網路SlurmDB中執行叢集和外部。

Slurmdbd在 AWS ParallelCluster v3.3.0 及更高版本中使用頭節點處Slurm理會計

在設定Slurm帳戶之前，您必須擁有使用mysql通訊協定的現有外部資料庫伺服器 and 資料庫。

若要使用配置Slurm會計 AWS ParallelCluster，您必須定義下列項目：

- 資料庫/Uri 中外部資料庫服务器的 [URI](#)。服务器必須存在且可從頭節點存取。

- 存取資料庫/[PasswordSecretArn](#)和 Database/中定義的外部[資料庫](#)的認證[UserName](#)。AWS ParallelCluster 使用此資訊來設定Slurm層次的帳戶，以及標頭節點上的slurmdbd服務。slurmdbd是管理叢集與資料庫伺服器之間通訊的守護程式。

若要逐步執行自學課程，請參閱[使用Slurm會計建立叢集](#)。

#### Note

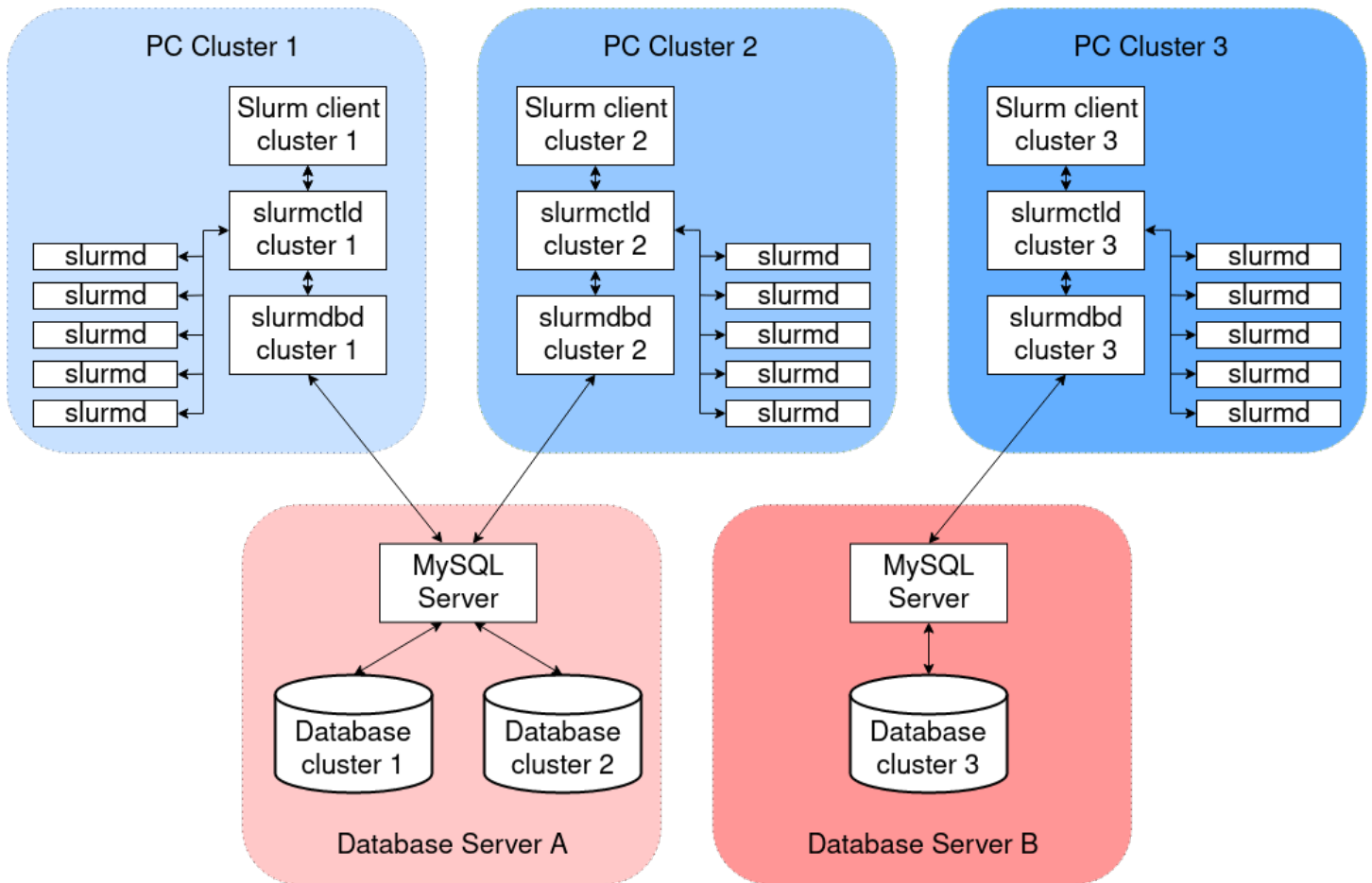
AWS ParallelCluster 在資料庫中將預設叢集使用者設定為資料庫管理員，以執行Slurm會計資料庫的基本啟動程序。Slurm AWS ParallelCluster 不會將任何其他使用者新增至會計資料庫。客戶負責管理資料庫中的會計實體。

AWS ParallelCluster 設定[slurmdbd](#)以確保叢集在Slurm資料庫伺服器上擁有自己的資料庫。同一個資料庫伺服器可以跨多個叢集使用，但每個叢集都有自己獨立的資料庫。AWS ParallelCluster 使用叢集名稱來定義slurmdbd組態檔[StorageLoc](#)參數中資料庫的名稱。請考慮下列情況。資料庫伺服器上存在的資料庫包含不對應至使用中叢集名稱的叢集名稱。在此情況下，您可以使用該叢集名稱建立新叢集，以對應至該資料庫。Slurm重複使用新叢集的資料庫。

#### Warning

- 我們不建議將多個叢集設定為一次使用相同的資料庫。這樣做可能會導致效能問題，甚至是資料庫死結的情況。
- 如果叢集的頭節點已啟用Slurm帳戶功能，建議您使用具有強大 CPU、更多記憶體和更高網路頻寬的執行個體類型。Slurm會計可以增加集群的頭節點應變。

在目前的 AWS ParallelCluster Slurm會計功能架構中，每個群集都有自己的slurmdbd守護進程實例，如下圖示例配置所示。



如果您要在叢集環境中新增自訂Slurm多叢集或同盟功能，則所有叢集都必須參考相同的slurmdbd執行個體。對於此替代方法，我們建議您在一個叢集上啟用 AWS ParallelCluster Slurm帳戶，並手動配置其他叢集以連接至第一slurmdbd個叢集上託管的叢集。

如果您使用的是 3.3.0 之前的版 AWS ParallelCluster 本，請參閱此 [HPC 部落格文章](#) 中所述的替代方法來實作Slurm會計。

## Slurm會計考量

### 不同 VPC 上的資料庫和叢集

若要啟用Slurm帳戶，需要資料庫伺服器做為slurmdbd精靈執行之讀取和寫入作業的後端。在建立或更新叢集以啟用Slurm帳戶之前，頭節點必須能夠連線到資料庫伺服器。

如果您需要在叢集使用的 VPC 以外的 VPC 上部署資料庫伺服器，請考慮下列事項：

- 若要啟用叢集端與資料庫伺服器之間的通訊，您必須設定兩個 VPC 之間的連線。slurmdbd如需詳細資訊，請參閱 Amazon Virtual Private Cloud 使用者 [指南中的 VPC 對等互連](#)。

- 您必須建立要連接至叢集 VPC 上之頭節點的安全性群組。對等兩個 VPC 之後，就可以使用資料庫端和叢集端安全性群組之間的交叉連結。如需詳細資訊，請參閱 Amazon Virtual Private Cloud 使用者指南中的[安全群組規則](#)。

## 設定和資料庫伺服器slurmdbd之間的 TLS 加密

如果伺服器支援 TLS 加密，則使用 AWS ParallelCluster 提供的預設Slurm帳戶組態slurmdbd建立與資料庫伺服器的 TLS 加密連線。AWS 資料庫服務 (例如 Amazon RDS) 和預設 Amazon Aurora 支援 TLS 加密。

您可以通過在數據庫服務器上設置require\_secure\_transport參數來在服務器端要求安全連接。這是在提供的 CloudFormation 模板中配置。

遵循安全性最佳作法，建議您同時在用slurmdbd戶端上啟用伺服器身分驗證。若要執行此操作，請在[StorageParameters](#)中配置slurmdbd.conf. 將伺服器 CA 憑證上傳至叢集的頭節點。接下來，將StorageParameters中的 [SSL\\_CA](#) 選項設定slurmdbd.conf為標頭節點上伺服器 CA 憑證的路徑。這樣做可以在slurmdbd側面啟用服務器身份驗證。進行這些變更之後，請重新啟動slurmdbd服務，以在啟用識別驗證的情況下重新建立與資料庫伺服器的連線。

## 更新資料庫認證

若要更新 [Database/UserName](#)或的值 [PasswordSecretArn](#)，您必須先停止運算叢集。假設存儲在密鑰中的 AWS Secrets Manager 秘密值已更改，並且其 ARN 不會更改。在此情況下，叢集不會自動將資料庫密碼更新為新值。若要更新新密碼值的叢集，請從 head 節點執行下列命令。

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

### Warning

為避免遺失帳戶資料，建議您僅在運算叢集停止時變更資料庫密碼。

## 資料庫監控

建議您啟用 AWS 資料庫服務的監視功能。如需詳細資訊，請參閱 [Amazon RDS 監控](#)或 [Amazon Aurora 監控](#)文件。

## Slurm組態自訂

開始於AWS ParallelCluster版本 3.6.0，您可以自定義slurm.conf Slurm組態中的AWS ParallelCluster叢集配置。

在叢集配置中，您可以自訂Slurm使用下列叢集配置設定的組態參數：

- 自訂Slurm整個叢集的參數，使用[SlurmSettings/CustomSlurmSettings](#)或[CustomSlurmSettingsIncludeFile](#)參數。AWS ParallelCluster如果同時指定兩者，則失敗。
- 自訂Slurm使用佇列的參數[SlurmQueues/CustomSlurmSettings](#)(對應至Slurm分區)。
- 自訂Slurm使用計算資源的參數[SlurmQueues/ComputeResources/CustomSlurmSettings](#)(對應至Slurm節點)。

### Slurm組態自訂限制和使用時的注意事項AWS ParallelCluster

- 對於CustomSlurmSettings和CustomSlurmSettingsIncludeFile設置，您只能指定和更新slurm.conf中包括的參數[Slurm版](#)這是由AWS ParallelCluster您用來配置叢集的版本。
- 如果您指定自訂Slurm組態中的任何CustomSlurmSettings參數，AWS ParallelCluster執行驗證檢查並防止設置或更新Slurm與衝突的組態參數AWS ParallelCluster邏輯。該Slurm已知會與之衝突的組態參數AWS ParallelCluster在拒絕清單中識別。拒絕清單可能會在未來變更AWS ParallelCluster版本，如果其他Slurm功能被添加。如需詳細資訊，請參閱[拒絕列出Slurm組態參數CustomSlurmSettings](#)。
- AWS ParallelCluster僅檢查參數是否在拒絕清單中。AWS ParallelCluster不驗證您的自定義Slurm配置參數語法或語義。您有責任驗證您的自定義Slurm組態參數。無效的自訂Slurm配置參數可能會導致Slurm可能導致叢集建立和更新失敗的協助程式失敗。
- 如果您指定自訂Slurm組態CustomSlurmSettingsIncludeFile,AWS ParallelCluster不執行任何驗證。
- 您可以更新CustomSlurmSettings和CustomSlurmSettingsIncludeFile無需停止和啟動計算叢集。在這種情況下，AWS ParallelCluster重新啟動slurmctld守護進程並運行scontrol reconfigure指令。

一些Slurm在整個叢集中註冊變更之前，配置參數可能需要不同的作業。例如，他們可能需要重新啟動叢集中的所有精靈。您有責任核實是否AWS ParallelCluster操作足以傳播您的自定義Slurm更新期間的組態參數設定。如果你發現AWS ParallelCluster作業不足，您有責任提供傳播更新設定所需的其他動作，如[Slurm文件](#)。



## 拒絕列出Slurm組態參數CustomSlurmSettings

下表列出了具有AWS ParallelCluster拒絕其使用的版本，從版本 3.6.0 開始。CustomSlurmSettings不支援AWS ParallelCluster版本比 3.6.0 版更早的版本。

叢集層級拒絕列出的參數：

Slurm 參數	拒絕上市AWS ParallelCluster版本
CommunicationParameters	3.6.0
Epilog	3.6.0
GresTypes	3.6.0
LaunchParameters	3.6.0
序言	3.6.0
ReconfigFlags	3.6.0
ResumeFailProgram	3.6.0
ResumeProgram	3.6.0
ResumeTimeout	3.6.0
SlurmctldHost	3.6.0
SlurmctldLogFile	3.6.0
SlurmctldParameters	3.6.0
SlurmdLogfile	3.6.0
SlurmUser	3.6.0
SuspendExcNodes	3.6.0
SuspendProgram	3.6.0
SuspendTime	3.6.0

Slurm 參數	拒絕上市AWS ParallelCluster版本
TaskPlugin	3.6.0
TreeWidth	3.6.0

在叢集層級拒絕列出的參數 [本地Slurm會計整合](#) 在叢集配置中配置：

Slurm 參數	拒絕上市AWS ParallelCluster版本
AccountingStorageType	3.6.0
AccountingStorageHost	3.6.0
AccountingStoragePort	3.6.0
AccountingStorageUser	3.6.0
JobAcctGatherType	3.6.0

在佇列 (分割區) 層級拒絕列出的參數，適用於所管理的佇列AWS ParallelCluster:

Slurm 參數	拒絕上市AWS ParallelCluster版本
節點	3.6.0
PartitionName	3.6.0
ResumeTimeout	3.6.0
州	3.6.0
SuspendTime	3.6.0

計算資源 (節點) 層級拒絕列出的參數 (由管理的計算資源)AWS ParallelCluster:

Slurm 參數	拒絕上市AWS ParallelCluster版本和更高版本
CPU	3.6.0

Slurm 參數	拒絕上市AWS ParallelCluster版本和更高版本
功能	3.6.0
格雷斯	3.6.0
NodeAddr	3.6.0
NodeHostname	3.6.0
NodeName	3.6.0
Weight	3.7.0

## Slurmprolog 與 epilog

從 AWS ParallelCluster 版本 3.6.0 開始，部署的Slurm組態包 AWS ParallelCluster 包括Prolog和Epilog組態參數：

```
# PROLOG AND EPILOG
Prolog=/opt/slurm/etc/scripts/prolog.d/*
Epilog=/opt/slurm/etc/scripts/epilog.d/*
SchedulerParameters=nohold_on_prolog_fail
BatchStartTimeout=180
```

有關更多信息，請參閱文檔中的 [Prolog 和 Epilog 指南](#)。Slurm

AWS ParallelCluster 包括以下序言和 epilog 腳本：

- 90\_plcluster\_health\_check\_manager ( 在Prolog文件夾中 )
- 90\_pcluster\_noop ( 在Epilog文件夾中 )

### Note

Prolog和資Epilog料夾必須至少包含一個檔案。

您可以通過將自己的自定義prolog或epilog腳本添加到相應的Prolog和Epilog文件夾中來使用它們。

**⚠ Warning**

Slurm以相反的字母順序執行資料夾中的每個指令碼。

prolog和epilog指令碼的執行時間持續時間會影響執行工作所需的時間。執行多個或長時間執行的prolog指令碼時，請更新BatchStartTimeout組態設定。預設值為 3 分鐘。

如果您使用的是自訂prolog和epilog指令碼，請在相應的Prolog和Epilog資料夾中找出指令碼。建議您保留在每個自訂90\_plcluster\_health\_check\_manager指令碼之前執行的指令碼。如需更多詳細資訊，請參閱 [Slurm組態自訂](#)。

## 叢集容量大小和更新

叢集的容量是由叢集可擴充的運算節點數量所定義。運算節點由組 AWS ParallelCluster 態中運算資源中定義的 Amazon EC2 執行個體提供支援([Scheduling/SlurmQueues/ComputeResources](#))，並組織成佇列([Scheduling/SlurmQueues](#))，以對應 1:1 到Slurm分區。

在計算資源中，您可以設定叢集中必須始終保持執行的最小計算節點 (執行個體[MinCount](#)) 數目，以及運算資源可擴充至 ([MaxCount3](#)) 的執行個體數目上限。

在叢集建立時或叢集更新時，MinCount針對叢集中定義的每個運算資源 ([Scheduling/SlurmQueues/ ComputeResources](#)) AWS ParallelCluster 啟動所設定數量的 Amazon EC2 執行個體。為了涵蓋叢集中運算資源的最小節點而啟動的執行個體稱為靜態節點。一旦啟動，靜態節點就應該在叢集中是永久性的，除非發生特定的事件或條件，否則它們不會被系統終止。例如，此類事件包括Slurm或 Amazon EC2 運作狀態檢查失敗，以及將Slurm節點狀態變更為「排出」或「關閉」。

Amazon EC2 執行個體的範圍1為 'MaxCount - MinCount' (MaxCount 減號 MinCount)，隨需啟動以處理叢集增加的負載) 稱為動態節點。它們的性質是短暫的，它們被啟動以提供待處理的工作，並且一旦它們[Scheduling/SlurmSettings/ScaledownIdletime](#)在叢集配置中定義的一段時間內保持閒置狀態 (預設值：10 分鐘)，就會終止。

靜態節點和動態節點符合下列命名結構描述：

- 靜態節點<Queue/Name>-st-<ComputeResource/Name>-<num>，其中 <num> = 1..ComputeResource/MinCount
- 動態節點<Queue/Name>-dy-<ComputeResource/Name>-<num>，其中 <num> = 1.. (ComputeResource/MaxCount - ComputeResource/MinCount)

例如，給出以下 AWS ParallelCluster 配置：

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

下面的節點將被定義 Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

當計算資源具有時 `MinCount == MaxCount`，所有對應的計算節點都將是靜態的，並且所有執行個體都會在叢集建立/更新時啟動，並保持正常運作。例如：

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

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
```

```
queue1*    up    infinite    100    idle queue1-st-c5xlarge-[1-100]
```

## 叢集容量更新

叢集容量的更新包括新增或移除佇列、計算資源或變更計算資源。MinCount/MaxCount從 AWS ParallelCluster 版本 3.9.0 開始，減少佇列的大小需要停止運算叢集或 [QueueUpdateStrategy](#) 設定為「終止」，才能進行叢集更新。在下列情況下，不需要停止運算叢集或設定 [QueueUpdateStrategy](#) 為「終止」：

- 將新佇列新增至排程/[SlurmQueues](#)
- 將新的計算資源新增 Scheduling/SlurmQueues/[ComputeResources](#) 至佇列
- 增加 [MaxCount](#) 運算資源
- 增加 MinCount 運算資源，並增加 MaxCount 至少相同數量的相同運算資源

## 考量與限制

本節旨在概述調整叢集容量大小時應考慮的任何重要因素、限制或限制。

- 從名稱 <Queue/Name> - \* 為靜態和動態的 Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues 所有運算節點移除佇列時，將從 Slurm 組態中移除，而對應的 Amazon EC2 執行個體也會終止。
- Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues-ComputeResources> ComputeResources 從佇列中移除運算資源時，所有具有靜態和動態名稱 <Queue/Name> - \* - <ComputeResource/Name> - \* 的運算節點都會從 Slurm 組態中移除，而對應的 Amazon EC2 執行個體也會終止。

當更改計算資源的 MinCount 參數時，我們可以區分兩種不同的場景，如果 MaxCount 保持等於 MinCount ( 僅靜態容量 )，以及 MaxCount 是否大於 MinCount ( 混合靜態和動態容量 )。

## 僅限靜態節點的容量變更

- 如果在增加 MinCount (和 MaxCount) 時  $\text{MinCount} == \text{MaxCount}$ ，將透過將靜態節點數量擴展到的新值來設定叢集，`MinCount<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` 且系統會繼續嘗試啟動 Amazon EC2 執行個體以滿足新的所需靜態容量。

- 如果  $\text{MinCount} == \text{MaxCount}$  在減少  $\text{MinCount}$  (和  $\text{MaxCount}$ ) 數量  $N$  時，將透過移除最後  $N$  個靜態節點來設定叢集 `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`]，系統將終止對應的 Amazon EC2 執行個體。

- 初始狀態  $\text{MinCount} = \text{MaxCount} = 100$

- 

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

- -30在  $\text{MinCount}$  和更新  $\text{MaxCount}$ :  $\text{MinCount} = \text{MaxCount} = 70$

- 

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

## 混合節點的容量變更

如果  $\text{MinCount} < \text{MaxCount}$  增加  $\text{MinCount}$   $N$  量 (假設  $\text{MaxCount}$  將保持不變)，則會將數量靜態節點擴充到新值  $\text{MinCount}$  ( $\text{old\_MinCount} + N$ ) 來設定叢集：`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` 系統將繼續嘗試啟動 Amazon EC2 執行個體以滿足新的必要靜態容量。此外，為了承受運算資源的  $\text{MaxCount}$  容量，透過移除最後  $N$  個動態節點來更新叢集組態：`<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>`] 系統將終止對應的 Amazon EC2 執行個體。

- 初始狀態： $\text{MinCount} = 100$ ;  $\text{MaxCount} = 150$

- 

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite    50  idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite   100  idle queue1-st-c5xlarge-[1-100]
```

- 更新至 MinCount : MinCount = 130 (MaxCount = 150)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

如果在增加MinCount且數量相同MaxCount的 N 時  $\text{MinCount} < \text{MaxCount}$ ，將透過將數量靜態節點擴展到  $\text{MinCount} (\text{old\_MinCount} + N)$  的新值來設定叢集：`<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N>`系統將繼續嘗試啟動 Amazon EC2 執行個體以滿足新的所需靜態容量。此外，不會對動態節點的數量進行任何更改以兌現新節點

MaxCount 值。

- 初始狀態 : MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 更新至 MinCount : MinCount = 130 (MaxCount = 180)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

如果  $\text{MinCount} < \text{MaxCount}$  在減少 MinCount N 量時 (假設保持不變)，則 MaxCount 會透過移除最後 N 個靜態節點靜態節點來設定叢集 `<Queue/Name>-st-<ComputeResource/Name>-[<old\_MinCount - N>...<old\_MinCount>`，系統將終止對應的 Amazon EC2 執行個體。



此外，為了承受運算資源的MaxCount容量，透過擴充動態節點的數量來更新叢集組態以填補空白。MaxCount - new\_MinCount: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new\_MinCount>]在這種情況下，由於這些節點是動態節點，因此除非排程器在新節點上有待處理的工作，否則不會啟動新的 Amazon EC2 執行個體。

- 初始狀態 : MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 在上的 -30 更新 MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   80    idle~ queue1-dy-c5xlarge-[1-80]
queue1*   up    infinite   70    idle queue1-st-c5xlarge-[1-70]
```

如果MinCount < MaxCount在減少MinCount且數MaxCount量相同 N 時，將透過移除最後 N 個靜態節點來設定叢集<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount - N>...<oldMinCount>]，系統將終止對應的 Amazon EC2 執行個體。

此外，不會對動態節點的數量進行任何更改以兌現新MaxCount值。

- 初始狀態 : MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 在上的 -30 更新 MinCount : MinCount = 70 (MaxCount = 120)

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   80    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite   70    idle  queue1-st-c5xlarge-[1-70]
```

如果  $\text{MinCount} < \text{MaxCount}$  在減少  $\text{MaxCount}$   $N$  量時 (假設  $\text{MinCount}$  將保持不變), 則會透過移除最後  $N$  個動態節點來設定叢集,  $\langle \text{Queue/Name} \rangle\text{-dy-}\langle \text{ComputeResource/Name} \rangle\text{-}\langle \text{old\_MaxCount} - N \dots \langle \text{oldMaxCount} \rangle ]$  而在執行的情況下, 系統將終止對應的 Amazon EC2 執行個體。預期不會對靜態節點造成任何影響。

- 初始狀態 :  $\text{MinCount} = 100$ ;  $\text{MaxCount} = 150$

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle  queue1-st-c5xlarge-[1-100]
```

- 在上的 -30 更新  $\text{MaxCount}$  :  $\text{MinCount} = 100$  ( $\text{MaxCount} = 120$ )

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up    infinite  100    idle  queue1-st-c5xlarge-[1-100]
```

## 對工作的影響

在移除節點且 Amazon EC2 執行個體終止的所有情況下, 除非沒有其他節點滿足任務要求, 否則在已移除節點上執行的 sbatch 任務都會重新排入佇列。在最後一種情況下, 工作將失敗, 狀態為 `NODE_FAIL` 並從隊列中消失; 如果是這種情況, 則需要手動重新提交。

如果您打算執行叢集調整大小更新，則可以防止工作在計劃更新期間移除的節點中執行。這可以通過設置要在維護中刪除的節點來實現。請注意，在維護中設定節點不會影響最終已在節點中執行的工作。

假設使用計劃的叢集調整大小更新，您將刪除節點 `queue-st-computeresource-[9-10]`。您可以使用下列命令建立 Slurm 保留區

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

這將建立在節點 `maint_for_update` 上命名的 Slurm 保留區 `queue-st-computeresource-[9-10]`。從建立保留區的時間開始，節點就不能再執行任何作業 `queue-st-computeresource-[9-10]`。請注意，保留區不會阻止最終在節點上配置工作 `queue-st-computeresource-[9-10]`。

叢集調整大小更新後，如果只在調整大小更新期間移除的節點上設定保 Slurm 留區，則維護保留區將會自動刪除。如果您在叢集調整大小更新之後仍然存在的節點上建立保 Slurm 留區，我們可能想要在執行調整大小更新之後移除節點上的維護保留區，方法是使用下列命令

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

[有關 Slurm 預訂的更多詳細信息，請在此處查看官方 ScheDMD 文檔。](#)

## 容量變更的叢集更新程序

排程器組態變更時，會在叢集更新程序期間執行下列步驟：

- 停止 AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- 從配置生成更新的 Slurm 分區配 AWS ParallelCluster 置
- 重新啟動 `slurmctld` (通過 Chef 服務配方完成)
- 檢查 `slurmctld` 狀態 (`systemctl is-active --quiet slurmctld.service`)
- 重新載入 Slurm 組 (`scontrol reconfigure`)
- 啟動 `clustermgtd` (`supervisorctl start clustermgtd`)

## AWS Batch (**awsbatch**)

如需 AWS Batch 的資訊，請參閱「[AWS Batch](#)」。如需說明文件，請參閱[AWS Batch 使用者指南](#)。

AWS ParallelCluster 適用於的 CLI 指令 AWS Batch

使用awsbatch排程器時，的 AWS ParallelCluster CLI 命令AWS Batch會自動安裝在AWS ParallelCluster頭節點中。CLI 使用 AWS Batch API 操作並允許以下操作：

- 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不支援的 GPU 工作AWS Batch。如需詳細資訊，請參閱 [GPU 工作](#)。

這個 CLI 是作為一個單獨的軟件包分發。如需詳細資訊，請參閱[排程器 Support](#)。

### 主題

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)
- [awsbqueues](#)
- [awsbhosts](#)

## awsbsub

將工作提交至叢集的工作佇列。

```
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 不支援的 GPU 工作 AWS Batch。如需詳細資訊，請參閱 [GPU 工作](#)。

## 定位引數

### **command**

提交工作 (指定的命令必須在計算執行個體上可用) 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。工作名稱可以包含字母 (大寫和小寫)、數字、連字號和底線，最長可包含 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，則會在使用者主目錄的 `job-<AWS_BATCH_JOB_ID>` 子資料夾中執行工作。您可以使用此參數或 `--parent-working-dir` 參數。

### **-pw *PARENT\_WORKING\_DIR*, --parent-working-dir *PARENT\_WORKING\_DIR***

指定工作目錄的父資料夾。如果未指定父工作目錄，它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***

指定要保留給容器的 vCPU 數目。與一起使用時 `-nodes`，它會識別每個節點的 vCPUs 數量。

預設：1

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

指定要提供給任務的記憶體的限制 (以 MiB 為單位)。如果您的工作嘗試超過此處指定的記憶體限制，工作便會結束。

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，'all' 環境變數的列表不包括 `-env-blacklist` 參數中列出的變數或以 `PCLUSTER_*` 或 `AWS_*` 前綴開頭的變數。

**-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***

指定將工作移至 `RUNNABLE` 狀態的次數。您可以指定嘗試 1 至 10 次。如果嘗試次數大於 1，則工作失敗時會重試，直到工作移至指定次數的 `RUNNABLE` 狀態為止。

預設：1

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

指定持續時間 (以秒為單位) (從作業嘗試的時間 `startedAt` 戳記測量)，如果工作尚未完成，則 AWS Batch 會終止工作。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 `parallel` 提交。

**Note**

當 [Scheduler/AwsBatchQueues/CapacityType](#) 參數設定為 `SPOT`，不支援多節點 `parallel` 工作。此外，您的帳戶中必須有 `AWSBatchQueuesRoleForEC2Spot` 服務連結角色。您可以使用下列 AWS CLI 命令建立此角色：

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

如需詳細資訊，請參閱 [Amazon 彈性運算雲端 Linux 執行個體使用者指南中的 Spot 執行個體請求的服務連結角色](#)。

### **-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，而不必指定陣列作業的工作 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。<string> 這個參數的語法是「jobId =<string>, 類型 =;...」。

## **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***

指定以空格分隔的清單，其中列出要在輸出中顯示的任務 ID。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***

取得任務輸出的第一個 *HEAD* 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***

指定以空格分隔的清單，其中列出要取消或終止的任務 ID。

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 ... ]]
```

位置引數

### *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。

選用引數

**-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*

指定空格分隔的執行個體 ID 清單。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

## 共用儲存

AWS ParallelCluster [支援使用 Amazon EBS、FSx 用於 ONTAP，以及用於 OpenZFS 共享儲存磁碟區的 FSx、Amazon EFS 和 FSx for Lustre 共享儲存檔案系統](#) 或 [檔案快取](#)。我們建議您遵循 [AWS 構建良好的架構可靠性支柱](#) 指引，並備份磁碟區和檔案系統。

選取符合 HPC 應用程式 I/O 需求的儲存系統。您可以根據您的特定使用案例來最佳化每個檔案系統。如需詳細資訊，請參閱 [儲存選項概觀](#)。

Amazon EBS 磁碟區會連接至主節點，並透過 NFS 與運算節點共用。此選項具有成本效益，但隨著儲存需求的擴充，效能取決於主節點資源。隨著叢集中增加了更多的運算節點，且輸送量需求增加，這可能會成為瓶頸。

Amazon EFS 檔案系統會隨儲存需求變更而擴展。您可以針對各種使用案例設定這些檔案系統。使用 Amazon EFS 檔案系統在叢集上執行並行化和延遲敏感應用程式。

FSx for Lustre 檔案系統可以處理大量資料集，速度高達每秒數百 GB 的輸送量、數百萬 IOPS 以及低於一毫秒的延遲。針對要求嚴苛的高效能運算環境，請將 FSx 用於 Lustre 檔案系統。

在中 [SharedStorage 區段](#)，您可以定義外部儲存或 AWS ParallelCluster 受管理的儲存：

- 外部儲存是指您管理的現有磁碟區或檔案系統。AWS ParallelCluster 不會建立或刪除此儲存空間。
- 受管理儲存是指 AWS ParallelCluster 建立並可刪除的磁碟區或檔案系統。

### 外接式儲存

您可 AWS ParallelCluster 以配置在建立或更新叢集時將外部儲存裝置連接至叢集。同樣地，您可以將其設定為在刪除或更新叢集時將外部儲存裝置從叢集中分離。您的資料會保留下來，您可以將其用於叢集生命週期以外的長期永久共用儲存。

#### Note

3.8 之 AWS ParallelCluster 前的版本不允許將外部管理的檔案系統掛載於 /home。從 3.8 版開始，AWS ParallelCluster 允許您用 /home 作外部託管文件系統的掛載點。您可以將外部管理的檔案系統掛載到 /home 中，方法是指定 /home 為 [MountDir](#) 參數的 [SharedStorage 區段](#) 值。Amazon 檔案快取不適合用作系統 /home 目錄，因此目前不支援掛載 /home。

在[SharedStorageType](#)組態選項下指定/home目錄時，將會覆寫，這表示[SharedStorage 區段](#)將改用下的設定。[SharedStorage 區段](#)

將外部檔案系統掛載到/home目錄時，將 head 節點的/home內容 AWS ParallelCluster 複製到外部文件系統，而不會覆蓋外部存儲上的現有文件。這包括為預設使用者傳輸叢集的 SSH 金鑰 (如果外部檔案系統不存在)。如需詳細資訊，請參閱[AWS ParallelCluster 共用儲存體考量](#)。

## AWS ParallelCluster 受管理儲存

AWS ParallelCluster 根據預設，受管理的儲存體取決於組態中叢集的生命週期。依預設[SharedStorageDeletionPolicy](#)設，組態參數Delete設定為。

根據預設，如果下列其中一項成立，則會刪除 AWS ParallelCluster 受管理的檔案系統或磁碟區及其資料。

- 您刪除叢集。
- 您變更受管理的共用儲存裝置組態Name。
- 您可以從組態中移除受管理的共用儲存區。

設定DeletionPolicy為可Retain保留受管理的共用檔案系統或磁碟區和資料。我們建議您定期備份數據，以避免數據丟失。您可以用[AWS Backup](#)來集中管理所有儲存空間選項的備份。

您可以使用組態設定移除生命週期相依性。如需詳細資訊，請參閱 [將 AWS ParallelCluster 受管儲存轉換為外部儲存](#)。

如需共用儲存配額的相關資訊，請參閱[共用儲存空間的配額](#)。

如需共用儲存裝置和切換至新 AWS ParallelCluster 版本的詳細資訊，請參閱[最佳做法：將叢集移至新的次要版本或 AWS ParallelCluster 修補程式版本](#)。

您可 AWS ParallelCluster 以配置在建立或更新叢集時將外部儲存裝置連接至叢集。同樣地，您可以將其設定為在刪除或更新叢集時將外部儲存裝置從叢集中分離。您的資料會被保留，您可以將其用於獨立於叢集生命週期的長期永久共用儲存解決方案。

根據預設，受管理的儲存體取決於叢集的生命週期。您可以使用中所述的組態設定來移除相依性[將 AWS ParallelCluster 受管儲存轉換為外部儲存](#)。

透過特定設定，您可以針對使用案例最佳化每個受支援的儲存解決方案。

如需共用儲存配額，請參閱[共用儲存空間的配額](#)。

如需共用儲存裝置和切換至新 AWS ParallelCluster 版本的詳細資訊，請參閱[最佳做法：將叢集移至新的次要版本或 AWS ParallelCluster 修補程式版本](#)。

## 主題

- [設定共用儲存](#)
- [使用中的共用儲存裝置 AWS ParallelCluster](#)
- [共用儲存空間的配額](#)

## 設定共用儲存

瞭解可用來定義叢集共用儲存裝置的組態設定。

## 主題

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx for Lustre](#)
- [為 ONTAP 設定 FSx、用於 OpenZF 的 FSx，以及檔案快取共用儲存裝置](#)

## Amazon Elastic Block Store

若要將現有的外部 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 [EbsSettings/VolumeId](#)。

如果未指定 [VolumeId](#)，依預設，AWS ParallelCluster 會在叢集建立 [EbsSettings](#) 時建立受管理的 EBS 磁碟區。AWS ParallelCluster 刪除叢集或磁碟區從叢集配置中移除時，也會刪除磁碟區和資料。

對於受 AWS ParallelCluster 管理的 EBS 磁碟區，您可以在刪除叢集或 Snapshot 從叢集配置中移除磁碟區時 Retain，使用 [EbsSettings/DeletionPolicy](#) 來指示 AWS ParallelCluster、或磁碟區。DeleteDeletionPolicy 預設會設定為 Delete。

### Warning

對於 AWS ParallelCluster 受管理的共用儲存裝置，DeletionPolicyDelete 預設為。這表示，如果下列其中一項成立，則會刪除受管理磁碟區及其資料：

- 您刪除叢集。

- 您變更受管理的共用儲存區組態 [SharedStorage/Name](#)。
- 您可以從組態中移除受管理的共用儲存區。

建議您定期使用快照備份共用儲存裝置，以避免資料遺失。如需有關 Amazon EBS 快照的詳細資訊，請參閱 [Amazon 彈性運算雲端 Linux 執行個體使用者指南中的 Amazon EBS 快照](#)。若要了解如何管理資料 [AWS Backup](#) AWS 服務，請參閱 [AWS Backup 開發人員指南中的備份](#)。

## Amazon Elastic File System

若要將現有的外部 Amazon EFS 檔案系統用於叢集生命週期以外的長期永久儲存 [FileSystemId](#)，請在 AWS ParallelCluster 建立叢集 [EfsSettings](#) 時指定 [EfsSettings/](#) 建立受管 Amazon EFS 檔案系統。AWS ParallelCluster 刪除叢集或從叢集配置中移除檔案系統時，也會刪除檔案系統和資料。

對於受 AWS ParallelCluster 管 Amazon EFS 檔案系統，您可以使用 [EfsSettings/DeletionPolicy](#) 指示刪除叢集 `DeleteRetain` 或 AWS ParallelCluster 何時刪除叢集，或從叢集組態中移除檔案系統時。DeletionPolicy 預設會設定為 `Delete`。

### Warning

對於 AWS ParallelCluster 受管理的共用儲存裝置，DeletionPolicy `Delete` 預設為。這表示，如果下列其中一項成立，則會刪除受管理的檔案系統及其資料：

- 您刪除叢集。
- 您變更受管理的共用儲存區組態 [SharedStorage/Name](#)。
- 您可以從組態中移除受管理的共用儲存區。

我們建議您定期備份共用儲存裝置，以避免資料遺失。如需有關如何 [備份個別 Amazon EFS 磁碟區的詳細資訊](#)，請參閱 [Amazon 彈性檔案系統使用者指南中的備份 Amazon EFS 檔案系統](#)。若要了解如何管理資料 [AWS Backup](#) AWS 服務，請參閱 [AWS Backup 開發人員指南中的備份](#)。

## Amazon FSx for Lustre

若要將現有的外部 FSx for Lustre 檔案系統用於叢集生命週期以外的長期永久儲存，請指定 [FsxLustreSettingsFileSystemId](#)。

如果未指定 [FsxLustreSettings/FileSystemId](#)，則依預設，AWS ParallelCluster 會在建立叢集 [FsxLustreSettings](#) 時為 Lustre 檔案系統建立受管理的 FSx。AWS ParallelCluster 刪除叢集或從叢集配置中移除檔案系統時，也會刪除檔案系統和資料。

對於 Lustre 的 AWS ParallelCluster 受管理 FSx 檔案系統，您可以使用 [FsxLustreSettings/DeletionPolicy](#) 指示 AWS ParallelCluster 刪除叢集 Delete 或 Retain 從叢集配置中移除檔案系統時指示或檔案系統。DeletionPolicy 預設會設定為 Delete。

#### Warning

對於 AWS ParallelCluster 受管理的共用儲存裝置，DeletionPolicy Delete 預設為。這表示，如果下列其中一項成立，則會刪除受管理的檔案系統及其資料：

- 您刪除叢集。
- 您變更受管理的共用儲存區組態 [SharedStorage/Name](#)。
- 您可以從組態中移除受管理的共用儲存區。

我們建議您定期備份共用儲存裝置，以避免資料遺失。您可以使用 [SharedStorage/FsxLustreSettings/AutomaticBackupRetentionDays](#) 和定義叢集中的備份 [DailyAutomaticBackupStartTime](#)。若要了解如何管理資料 [AWS Backup](#) AWS 服務，請參閱 [AWS Backup 開發人員指南](#) 中的備份。

## 為 ONTAP 設定 FSx、用於 OpenZF 的 FSx，以及檔案快取共用儲存裝置

對於 ONTAP 的 FSx、OpenZFS 的 FSx 和檔案快取，您可以使用 [FsxOntapSettings/VolumeId](#)、[FsxOpenZfsSettings/](#)和 [FileCacheSettings/FileCacheId](#) 來指定叢集的外部現有磁碟區或檔案快取。 [VolumeId](#)

AWS ParallelCluster 不支援受管理的共用儲存裝置，適用於 ONTAP 的 FSx、用於 OpenZF 的 FSx 和檔案快取。

## 使用中的共用儲存裝置 AWS ParallelCluster

瞭解如何使用 AWS ParallelCluster 和共用儲存裝置。

### 主題

- [AWS ParallelCluster 共用儲存體考量](#)

- [將 AWS ParallelCluster 受管儲存轉換為外部儲存](#)

## AWS ParallelCluster 共用儲存體考量

使用中的共用儲存時，請考慮下列事項 AWS ParallelCluster。

- 使用[AWS Backup](#)或其他方法來備份您的檔案系統資料，以管理所有儲存系統的備份。
- 若要新增共用儲存裝置，請將共用儲存區段新增至組態檔案，然後建立或更新叢集。
- 若要移除共用儲存裝置，請從配置檔案中移除共用儲存區段，然後更新叢集。
- 若要以新的 AWS ParallelCluster 受管理的儲存區取代現有的受管理共用儲存裝置，請變更 [SharedStorage](#) 的值，[Name](#) 然後更新叢集。

### Warning

依預設，當您使用新Name參數執行叢集更新時，會刪除現有 AWS ParallelCluster 受管理的儲存區和資料。如果您需要變更Name和保留現有的受管理共用儲存體資料，請確定在更新叢集之前DeletionPolicy將資料設定為Retain或備份資料。

- 如果您沒有備份 AWS ParallelCluster 受管理的儲存體資料Delete，DeletionPolicy當您的叢集遭到刪除，或從叢集配置中移除受管理的儲存裝置並更新叢集時，您的資料會被刪除。
- 如果您沒有備份 AWS ParallelCluster 受管理的儲存資料Retain，DeletionPolicy您的檔案系統會在刪除叢集之前中斷連接，並且可以重新連接到另一個叢集做為外部檔案系統。您的資料會被保留。
- 如果 AWS ParallelCluster 受管理的儲存裝置已從叢集配置中移除Retain，DeletionPolicy則可以將其作為外部檔案系統重新連接至叢集，並保留叢集資料。
- 從 3.4.0 AWS ParallelCluster 版開始，您可以透過設定 [SharedStorage/EfsSettings/EncryptionInTransit](#) 和 [IamAuthorization](#) 設定來增強 Amazon EFS 檔案系統掛載的安全性。
- 將外部檔案系統掛載到 /home 目錄時，會將 head 節點 /home 目錄的內容 AWS ParallelCluster 複製到外部檔案系統。它會複製 /home 目錄中的現有資料，而不會覆寫外部儲存裝置上的現有檔案或目錄。這包括群集的 SSH 密鑰，以防默認用戶不存在於外部文件系統中。因此，將相同外部檔案系統掛載到各自 /home 目錄的所有其他叢集，對於叢集的預設使用者，也會擁有相同的 SSH 金鑰。
- 在將相同外部檔案系統掛載至叢集 /home 目錄的多叢集環境中，只有當第一個叢集將外部檔案系統掛載至 /home 時，才會產生一次授與存取權限 (在頭節點上建立之運算節點的 SSH 金鑰)。AWS ParallelCluster所有其他叢集都使用相同的 SSH 金鑰。因此，任何擁有這些共用叢集之預設使用者的 SSH 金鑰的人都可以存取任何叢集。所有運算節點都允許使用最初產生的金鑰進行連線。



## 將 AWS ParallelCluster 受管儲存轉換為外部儲存

瞭解如何將 AWS ParallelCluster 受管儲存轉換為外部儲存裝置。

這些程序以下列範例組態檔案片段為基礎。

```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Delete
...
```

### 將 AWS ParallelCluster 受管儲存轉換為外部儲存

1. 在叢集配置檔案Retain中DeletionPolicy將設定為。

```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Retain
...
```

2. 若要設定DeletionPolicy變更，請執行下列命令。

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

3. 從叢集配置檔移除SharedStorage段落。

```
...
...
```

4. 若要將受管理的外部變更SharedStorage為外部SharedStorage並將其從叢集中斷連結，請執行下列命令。

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. 您的共用儲存裝置現在是外部的，而且已從叢集中分離。
6. 若要將外部檔案系統附加至原始叢集或其他叢集，請依照下列步驟執行。
  - a. 取得 Lustre 檔案系統識別碼的 FSx。
    - i. 若要使用 AWS CLI 執行下列命令，尋找名稱包含原始叢集名稱的檔案系統，並記下檔案系統 ID。

```
aws fsx describe-file-systems
```

- ii. 若要使用 AWS Management Console，請登入並瀏覽至 <https://console.aws.amazon.com/fsx/>。在檔案系統清單中，尋找名稱包含原始叢集名稱的檔案系統，並記下檔案系統 ID。
  - b. 更新檔案系統安全性群組規則，以提供檔案系統和叢集子網路的存取權。您可以在 Amazon FSx 主控台中找到檔案系統安全群組名稱和 ID。

將規則新增至檔案系統安全性群組，該群組允許來自頭節點和運算節點 IP CIDR 範圍或首碼的輸入和輸出 TCP 流量。指定輸入和輸出 TCP 流量的通訊埠 988、1021、1022 及 1023。

如需詳細資訊，請參閱第 2 版 AWS Command Line Interface 使用者指南中的 [SharedStorageFsxLustreSettings//FileSystemId](#) 和建立、設定和刪除 Amazon EC2 的 [安全群組](#)。

- c. 將 SharedStorage 區段新增至叢集配置。

```
...
- MountDir: /fsx
  Name: fsx-external
  StorageType: FsxLustre
  FsxLustreSettings:
    FileSystemId: fs-02e5b4b4abd62d51c
...
```

- d. 若要將外部共用儲存裝置新增至叢集，請執行下列命令。

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

## 共用儲存空間的配額

設定叢集SharedStorage以掛載現有的共用檔案儲存裝置，並根據下表所列的配額建立新的共用檔案儲存體。

### 每個叢集的掛載檔案儲存配額

| 檔案共用儲存體類型    | AWS ParallelCluster<br>受管理儲存 | 外接式儲存 | 總額淨額 |
|--------------|------------------------------|-------|------|
| Amazon EBS   | 5                            | 5     | 5    |
| 突襲           | 1                            | 0     | 1    |
| Amazon EFS   | 1                            | 20    | 21   |
| Amazon FSx † | 1 FSx for Lustre             | 20    | 21   |

#### Note

此配額表格已在 3.2.0 AWS ParallelCluster 版中新增。

† AWS ParallelCluster 僅支援掛載現有的 Amazon FSX，適用於 NetApp OpenZF 的 Amazon FSX 和檔案快取系統。它不支援建立適用於 ONTAP 的新 FSx、OpenZF 的 FSx，以及檔案快取系統。

#### Note

如果您使用 AWS Batch 作為排程器，則 FSx for Lustre 只能在叢集標頭節點上使用。檔案快取不支援 AWS Batch 排程器。

## AWS ParallelCluster 資源和標記

AWS ParallelCluster 您可以使用建立標籤來追蹤和管理您的 AWS ParallelCluster 資源。您可以定義要建立並傳播 AWS CloudFormation 至叢集配置檔中所有叢集資源 [Tags 區段](#) 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。

建立叢集時，叢集及其資源會以本節中定義的 AWS ParallelCluster 和 AWS 系統標籤加上標籤。

AWS ParallelCluster 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，請將 AWS 系統標記 AWS CloudFormation 套用至叢集執行個體。為了識別叢集的 Amazon EC2 啟動範本，Amazon EC2 會將系統標籤套用至執行個體。您可以使用這些標籤來檢視和管理 AWS ParallelCluster 源。

您無法修改 AWS 系統標籤。為了避免對 AWS ParallelCluster 功能造成影響，請勿修改 AWS ParallelCluster 標籤。

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

```
"aws:cloudformation:stack-name"="clustername"
```

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

```
"parallelcluster:cluster-name"="clustername"
```

您可以在的 Amazon EC2 部分中查看這些標籤 AWS Management Console。

### 檢視標籤

1. 在以下位置瀏覽 Amazon EC2 控制台 <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. 若要依執行處理檢視叢集標記，請在導覽窗格中選擇 [執行個體]
4. 選取叢集執行處理。
5. 在執行個體詳細資料中選擇管理標籤，然後檢視標籤。
6. 在執行個體詳細資料中選擇儲存空間標籤。
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「標記」標籤，然後檢視標籤。

### AWS ParallelCluster 頭節點實例標籤

| 金鑰                           | 標籤值                |
|------------------------------|--------------------|
| parallelcluster:cluster-name | <i>clustername</i> |
| Name                         | HeadNode           |

| 金鑰                            | 標籤值                                                                                                                                       |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| aws:ec2launchtemplate:id      | <i>lt-1234567890abcdef0</i>                                                                                                               |
| aws:ec2launchtemplate:version | <i>1</i>                                                                                                                                  |
| parallelcluster:node-type     | HeadNode                                                                                                                                  |
| aws:cloudformation:stack-name | <i>clustername</i>                                                                                                                        |
| aws:cloudformation:logical-id | HeadNode                                                                                                                                  |
| aws:cloudformation:stack-id   | arn:aws:cloudformation: <i>region-id</i> : <i>ACCOUNTID</i> :stack/ <i>clustername</i> / <i>1234abcd-12ab-12ab-12ab-1234567890abcdef0</i> |
| parallelcluster:version       | <i>3.7.0</i>                                                                                                                              |

#### AWS ParallelCluster 頭節點根卷標籤

| 標籤鍵                          | 標籤值                |
|------------------------------|--------------------|
| parallelcluster:cluster-name | <i>clustername</i> |
| parallelcluster:node-type    | HeadNode           |
| parallelcluster:version      | <i>3.7.0</i>       |

#### AWS ParallelCluster 計算節點例項標籤

| 金鑰                                    | 標籤值                          |
|---------------------------------------|------------------------------|
| parallelcluster:cluster-name          | <i>clustername</i>           |
| parallelcluster:compute-resource-name | <i>compute-resource-name</i> |
| aws:ec2launchtemplate:id              | <i>lt-1234567890abcdef0</i>  |

| 金鑰                            | 標籤值               |
|-------------------------------|-------------------|
| aws:ec2launchtemplate:version | <i>1</i>          |
| parallelcluster:node-type     | Compute           |
| parallelcluster:queue-name    | <i>queue-name</i> |
| parallelcluster:version       | <i>3.7.0</i>      |

### AWS ParallelCluster 計算節點根磁碟區標記

| 標籤鍵                                   | 標籤值                          |
|---------------------------------------|------------------------------|
| parallelcluster:cluster-name          | <i>clustername</i>           |
| parallelcluster:compute-resource-name | <i>compute-resource-name</i> |
| parallelcluster:node-type             | Compute                      |
| parallelcluster:queue-name            | <i>queue-name</i>            |
| parallelcluster:version               | <i>3.7.0</i>                 |

### AWS ParallelCluster 介面標籤

| 標籤鍵                | 標籤值  |
|--------------------|------|
| parallelcluster-ui | true |

## 監控 AWS ParallelCluster 和記錄

監控是維持其他 AWS 解決方案的可靠性、可用性和效能的 AWS ParallelCluster 重要組成部分。AWS 提供下列監控工具來監視 AWS ParallelCluster、在發生錯誤時回報，並在適當時自動採取行動：

- Amazon 會即時 CloudWatch 監控您的 AWS 資源和執行 AWS 的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您

可以 CloudWatch 追蹤 Amazon EC2 執行個體的 CPU 使用率或其他指標，並在需要時自動啟動新執行個體。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。

- Amazon CloudWatch 日誌可讓您從 Amazon EC2 執行個體和其他來源監控 CloudTrail、存放和存取日誌檔。CloudWatch 記錄檔可以監控記錄檔中的資訊，並在符合特定臨界值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南](#)。
- AWS CloudTrail 擷取您 AWS 帳戶 發出或代表發出的 API 呼叫和相關事件，並傳送日誌檔案至您指定的 Amazon S3 儲存貯體。您可以找出哪些使用者和帳戶呼叫 AWS、發出呼叫的來源 IP 地址，以及呼叫的發生時間。如需詳細資訊，請參閱 [AWS CloudTrail 使用者指南](#)。
- Amazon EventBridge 是一種無伺服器事件匯流排服務，可讓您輕鬆地將應用程式與各種來源的資料連接起來。EventBridge 從您自己的應用程式、Software-as-a 服務 (SaaS) 應用程式以及服務提供即時資料串流，並 AWS 將該資料路由到目標 (例如 Lambda)。這可讓您監控在服務中發生的事件，並建置事件導向的架構。如需詳細資訊，請參閱 [Amazon EventBridge 使用者指南](#)。

## 主題

- [與 Amazon Amazon Amazon CloudWatch 的整合](#)
- [Amazon CloudWatch 儀表](#)
- [針對叢集指標的 Amazon CloudWatch 警示](#)
- [AWS ParallelCluster 配置的日誌輪替](#)
- [pclusterCLI 記錄檔](#)
- [Amazon EC2 主控台輸出日誌](#)
- [擷取AWS ParallelCluster UI 和AWS ParallelCluster執行階段記錄](#)
- [擷取和保留記錄](#)

## 與 Amazon Amazon Amazon CloudWatch 的整合

如需有關 CloudWatch 日誌的詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南](#)。若要設定 CloudWatch 記錄檔整合，請參閱 [Monitoring](#) 章節。若要了解如何使用將自訂日誌附加到 CloudWatch 組態 append-config，請參閱 Amazon 使用 CloudWatch 者指南中的 [多個 CloudWatch 代理程式組態檔案](#)。

## 亞馬遜CloudWatch日誌群集日誌

系統會為每個叢集建立一個名稱 `/aws/parallelcluster/cluster-name-<timestamp>` (例如 `/aws/parallelcluster/testCluster-202202050215`) 的記錄群組。每個節點上的每個日誌 ( 或一組日誌 , 如果路徑包含 \* ) 都有一個名為的日誌流 `{hostname}.{instance_id}.{logIdentifier}`。 ( 例如 `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`。 ) 日誌資料CloudWatch由代理程式傳送至 , [CloudWatch代理程式](#) 會與所有叢集執root行個體一樣執行。

建立叢集時 , 會建立 Amazon Amazon CloudWatch 儀表板。此儀表板可讓您檢閱記錄中儲存的 CloudWatch記錄。如需詳細資訊 , 請參閱 [Amazon CloudWatch 儀表](#)。

此清單包含 log *Identifier* 和可用於平台、排程器和節點的記錄串流路徑。

可用於平台、排程器和節點的記錄串流

| 平台                               | 排程器           | 節點            | 日誌串流                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------|---------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| amazon<br>CENTO<br>紅帽子<br>ubuntu | awsbatc<br>泥漿 | HeadNc        | DCV-身份驗證器 : <code>/var/log/parallelcluster/parallelcluster_dcv_authenticator.log</code><br>dcv-ext-authenticator: <code>/var/log/parallelcluster/parallelcluster_dcv_connect.log</code><br>DCV-代理: <code>/var/log/dcv/agent.*.log</code><br>DC 交互作業階段: <code>/var/log/dcv/dcv-xsession.*.log</code><br>DCV-服務器 : <code>/var/log/dcv/server.log</code><br>dcv-session-launcher: <code>/var/log/dcv/sessionlauncher.log</code><br>XDCV : <code>/var/log/dcv/Xdcv.*.log</code><br>cfn-init : <code>/var/log/cfn-init.log</code><br>廚師客戶 : <code>/var/log/chef-client.log</code> |
| amazon<br>CENTO                  | awsbatc<br>泥漿 | Comput<br>eet | 雲初始化 : <code>/var/log/cloud-init.log</code><br>監督 : <code>/var/log/supervisord.log</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |



| 平台                               | 排程器 | 節點            | 日誌串流                                                                                                                                                                                    |
|----------------------------------|-----|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 紅帽子<br>ubuntu                    |     | HeadNode      |                                                                                                                                                                                         |
| amazon<br>CENTO<br>紅帽子<br>ubuntu | 泥漿  | Compute<br>et | cloud-init-output: /var/log/cloud-init-output.log<br>計算: /var/log/parallelcluster/computemgtd<br>泥漿: /var/log/slurmd.log<br>思羅姆_生理_癲癇: /var/log/parallelcluster/slurm_prolog_epilog.log |

| 平台     | 排程器      | 節點       | 日誌串流                                                               |
|--------|----------|----------|--------------------------------------------------------------------|
| amazon | 泥漿       | HeadNode | 社會科學部 : /var/log/sssds/sssds.log                                   |
| CENTOS |          |          | 網域預設值 : /var/log/sssds/sssds_default.log                           |
| 紅帽子    |          |          | 密鑰生成器 : /var/log/parallelcluster/pam_ssh_key_generator.log         |
| ubuntu |          |          | 叢集狀態 : /var/log/parallelcluster/clusterstatusmgt                   |
|        |          |          | 集群 : /var/log/parallelcluster/clustermgtd                          |
|        |          |          | 電腦輸出 : /var/log/parallelcluster/compute_console_output             |
|        |          |          | 思樂簡歷 : /var/log/parallelcluster/slurm_resume.log                   |
|        |          |          | 思盧姆暫停 : /var/log/parallelcluster/slurm_suspend.log                 |
|        |          |          | 思樂麥克頂級: /var/log/slurmctld.log                                     |
|        |          |          | 思樂隊狀態管理器 : /var/log/parallelcluster/slurm_fleet_status_manager.log |
| amazon | awsbatch | Compute  | 系統訊息 : /var/log/messages                                           |
| CENTOS | 泥漿       | HeadNode |                                                                    |
| 紅帽子    |          |          |                                                                    |
| ubuntu | awsbatch | Compute  | 系統日誌 : /var/log/syslog                                             |
|        | 泥漿       | HeadNode |                                                                    |

叢集中使用的工作會將達到RUNNING、SUCCEEDED或狀態的工作輸出AWS Batch儲存FAILED在CloudWatch記錄檔中。記錄群組為/aws/batch/job，記錄資料流名稱格式為`jobDefinitionName/default/ecs_task_id`。根據預設，這些記錄檔設定為不到期，但您可以修改保留期間。如需詳細資訊，請參閱《Amazon Logs User Guide》中的變更CloudWatch日誌中的CloudWatch日誌[資料保留期間](#)。

## 亞馬遜CloudWatch日誌構建映像日誌

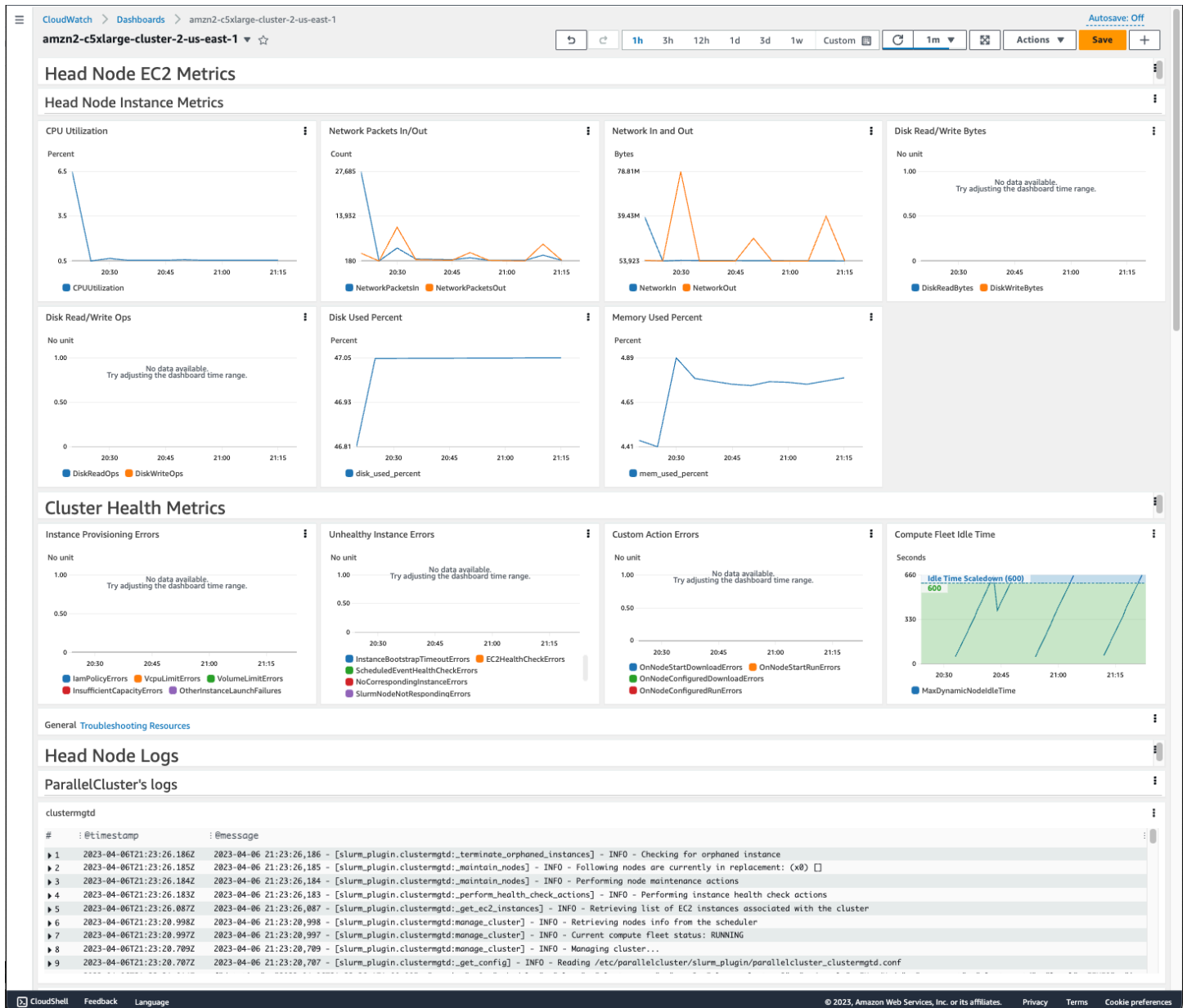
系統會為每個自訂組建映像建立一個記錄群組，名稱為/aws/imagebuilder/ParallelClusterImage-`<image-id>`。名稱為 `{pcluster-version}/1` 的唯一日誌流包含構建映像過程的輸出。

您可以使用 `image` 指令存取 `pcluster` 記錄檔。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 定制](#)。

## Amazon CloudWatch 儀表

建立叢集時會建立 Amazon CloudWatch 儀表板。這可讓您更輕鬆地監控叢集中的節點，以及檢視存放在 Amazon Logs 中的 CloudWatch 日誌。儀表板的名稱為 `ClusterName-Region`。 `ClusterName` 是叢集的名稱，而 `[##]` AWS 區域 是叢集所在。您可以在主控台中存取儀表板，也可以透過開啟來存取 `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`。

下圖顯示叢集的範例 CloudWatch 儀表板。



## 頭節點執行處理測量

儀表板的第一個區段會顯示主節點 Amazon EC2 指標的圖形。

如果您的叢集具有共用儲存裝置，下一節會顯示共用儲存體指標。

## 叢集 Health 指

如果您的叢集用 Slurm 於排程，叢集健康狀況指標圖表會顯示即時叢集運算節點錯誤。如需詳細資訊，請參閱 [疑難排解叢集健康度](#)。叢集健康狀態指標會從 3.6.0 AWS ParallelCluster 版開始新增至儀表板。

## 標頭節點記錄

最後一節列出了按日誌，排程器 AWS ParallelCluster 的日誌，NICE DCV 集成日誌和系統日誌分組的頭節點日誌。

如需 Amazon CloudWatch 儀表板的詳細資訊，請參閱 [Amazon 使用 CloudWatch 者指南中的使用 Amazon CloudWatch 儀表板](#)。

如果您不想創建 Amazon CloudWatch 儀表板，則可以通過將 [Monitoring//DashboardsCloudWatch](#) 設置 `Enabled` 為將其關閉 `false`。

### Note

如果停用 Amazon CloudWatch 儀表板的建立，也會停用叢集的 Amazon CloudWatch `disk_used_percent` 和 `memory_used_percent` 警示。如需詳細資訊，請參閱 [針對叢集指標的 Amazon CloudWatch 警示](#)。  
`disk_used_percent` 和 `memory_used_percent` 鬧鐘會從 3.6 AWS ParallelCluster 版開始新增。

## 針對叢集指標的 Amazon CloudWatch 警示

從 3.6 AWS ParallelCluster 版開始，您可以使用 Amazon CloudWatch 警示來設定叢集，以監控主節點。一個警報會監控根磁碟區 `disk_used_percent`。另一個警報會監控 `mem_used_percent` 度量。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的 CloudWatch 代理程式收集的 [指標](#)。

警報的名稱如下：

- `cluster-name_DiskAlarm_HeadNode`
- `cluster-name_MemAlarm_HeadNode`

`cluster-name`。

在導覽窗格中選擇 [警報]，存取 CloudWatch 主控台中的警報。下列影像顯示叢集的磁碟使用量警示和記憶體使用量警示。

CloudWatch > Alarms > test-disk-alarm-1\_DiskAlarm\_HeadNode

**Alarms (11)**

Any state

Any type

Any actions status

Hide Auto Scaling alarms

< 1 >

**test-disk-alarm-1\_DiskAlarm\_HeadNode**

Metric alarm

OK

test-disk-alarm-1\_MemAlarm\_HeadNode

Metric alarm

OK

mytest

Metric alarm

Insufficient data

**test-disk-alarm-1\_DiskAlarm\_HeadNode**

### Graph

**disk\_used\_percent**  OK

disk\_used\_percent > 90 for 1 datapoints within 1 minute

Percent

90.00

67.67

45.35

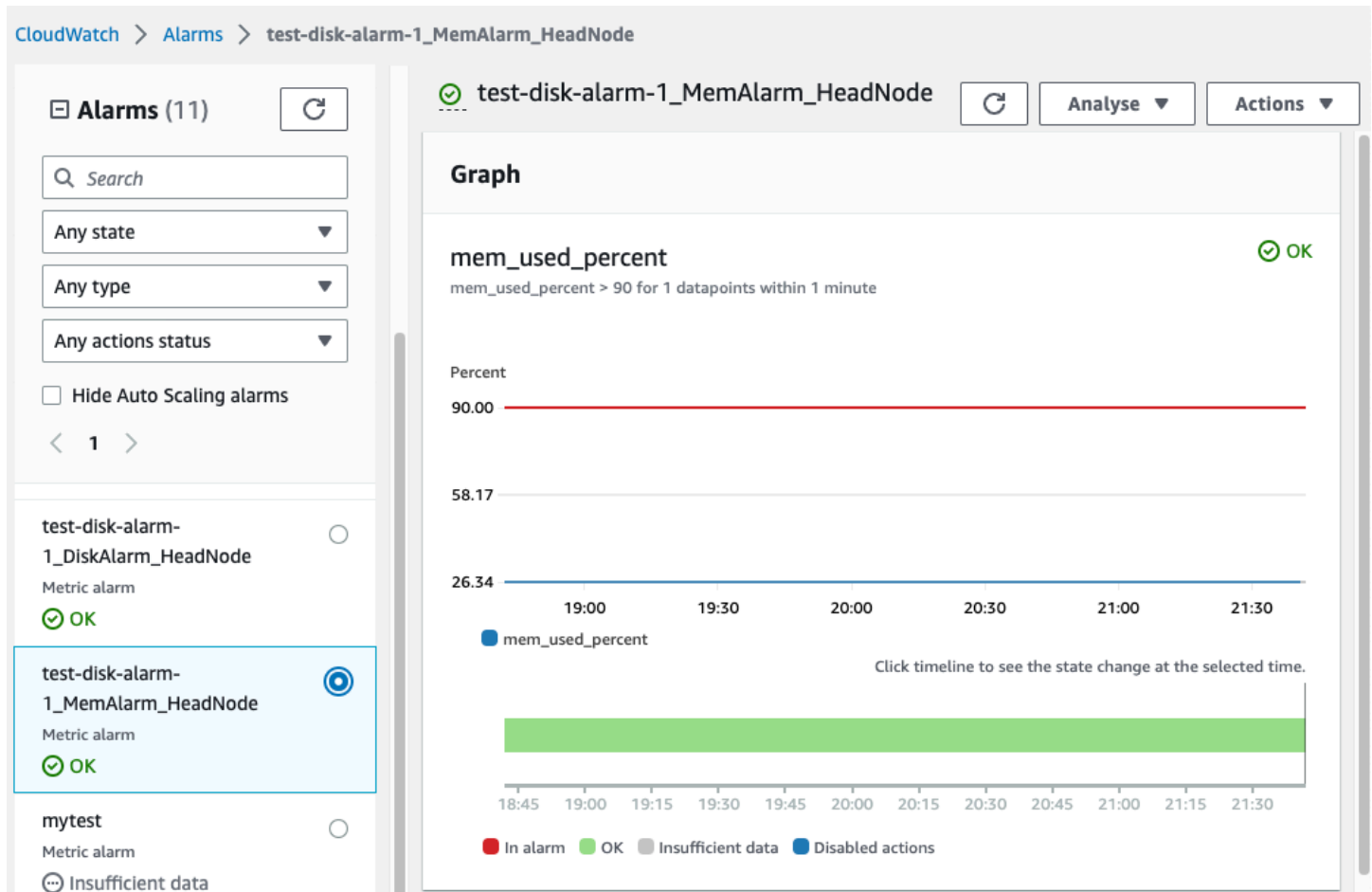
19:00 19:30 20:00 20:30 21:00 21:30

disk\_used\_percent

Click timeline to see the state change at the selected time.

18:45 19:00 19:15 19:30 19:45 20:00 20:15 20:30 20:45 21:00 21:15 21:30

In alarm  OK  Insufficient data  Disabled actions



磁碟使用率警示處於 1 個資料點的磁碟使用率百分比大於 90% 的ALARM狀態 (在 1 分鐘的時間內)。

記憶體使用量警示處於 1 個資料點的記憶體使用率百分比大於 90% 的ALARM狀態，在 1 分鐘的時間內。

### Note

AWS ParallelCluster 默認情況下不配置警報操作。如需如何設定警示動作 (例如傳送通知) 的相關資訊，請參閱[警示動作](#)。如需 Amazon CloudWatch 警示的詳細資訊，請參閱 [Amazon 使用 CloudWatch 者指南中的使用 Amazon CloudWatch 警示](#)。

如果您不想建立這些 Amazon CloudWatch 警報，請false在叢集組態中將 [Monitoring/Dashboards/CloudWatch/設定Enabled](#) 為來停用警示。這也會禁用亞馬遜 CloudWatch 儀表板的創建。如需詳細資訊，請參閱 [Amazon CloudWatch 儀表](#)。

**Note**

如果您停用 Amazon CloudWatch 儀表板的創建，也會停用 Amazon CloudWatch `disk_used_percent` 和叢集的 `memory_used_percent` 警示。

## AWS ParallelCluster 配置的日誌輪替

該 AWS ParallelCluster 記錄輪換組態位於 `/etc/logrotate.d/parallelcluster_*_log_rotation` 文件。當設定的記錄檔輪換時，目前的記錄檔內容會保留在單一備份中，而清空的記錄會繼續記錄。

每個設定的記錄只會維護 1 個備份。

AWS ParallelCluster 設定快速成長的記錄檔，以便在大小達到 50 MB 時進行輪換。快速成長的日誌與擴展和 Slurm，包括 `/var/log/parallelcluster/clustermgtd`、`/var/log/parallelcluster/slurm_resume.log`，以及 `/var/log/slurmctld.log`。

AWS ParallelCluster 設定成長緩慢的記錄檔，以便在大小達到 10 MB 時進行旋轉。

您可以檢視在叢集配置中定義的天數內保留的先前記錄 [Logs/CloudWatch/RetentionInDays](#) 設定與 CloudFormation 已啟用記錄。檢查 `RetentionInDays` 設定以查看您的使用案例是否需要增加天數。

AWS ParallelCluster 設定及輪換下列記錄檔：

### 標頭節點記錄

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cfn-init.log
/var/log/chef-client.log
/var/log/dcv/server.log
/var/log/dcv/sessionlauncher.log
/var/log/dcv/agent.*.log
/var/log/dcv/dcv-xsession.*.log
/var/log/dcv/Xdcv.*.log
/var/log/parallelcluster/pam_ssh_key_generator.log
/var/log/parallelcluster/clustermgtd
/var/log/parallelcluster/clusterstatusmgtd
/var/log/parallelcluster/slurm_fleet_status_manager.log
/var/log/parallelcluster/slurm_resume.log
```



```
/var/log/parallelcluster/slurm_suspend.log
/var/log/slurmctld.log
/var/log/slurmdbd.log
/var/log/parallelcluster/compute_console_output.log
```

## 計算節點記錄

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log
```

## 登入節點記錄

```
/var/log/cloud-init.log
/var/log/cloud-init.log
/var/log/cloud-init-output.log
/var/log/supervisord.log
/var/log/parallelcluster/pam_ssh_key_generator.log
```

## pclusterCLI 記錄檔

pclusterCLI 會將命令的記錄寫入中的 `pcluster.log`。#檔案/home/user/.parallelcluster/。

對於每個命令，日誌通常包括帶有輸入的命令，用於製作命令的 CLI API 版本的副本，響應以及信息和錯誤消息。對於創建和構建命令，日誌還包括配置文件，配置文件驗證操作，CloudFormation 模板和堆棧命令。

您可以使用這些記錄來驗證錯誤、輸入、版本和 pcluster CLI 命令。它們還可以作為執行命令的記錄。

## Amazon EC2 主控台輸出日誌

當 AWS ParallelCluster 偵測到靜態運算節點執行個體意外終止時，會在經過一段時間後嘗試從終止的節點執行個體擷取 Amazon EC2 主控台輸出。如此一來，如果運算節點無法與 Amazon 通訊 CloudWatch，仍可從主控台輸出擷取有關終止節點原因的實用疑難排解資訊。此控制台輸出記 `/var/log/parallelcluster/compute_console_output` 錄在頭節點的日誌中。[如需 Amazon EC2 主控台輸出的詳細資訊，請參閱 Amazon EC2 執行個體使用者指南中的執行個體主控台輸出。](#)

依預設，AWS ParallelCluster 只會從終止節點的範例子集擷取主控台輸出。這樣可防止叢集頭節點因大量終止而導致多個主控台輸出要求不堪重負。依預設，在終止偵測和主控台輸出擷取之間 AWS ParallelCluster 等待 5 分鐘，讓 Amazon EC2 有時間從節點擷取最終主控台輸出。

您可以在頭節點上的 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` 檔案中編輯樣本大小和等待時間參數值。

此功能已在 3.5.0 AWS ParallelCluster 版中添加。

## Amazon EC2 控制台輸出參數

您可以在頭節點上的 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` 檔案中編輯下列 Amazon EC2 主控台輸出參數的值。

### **compute\_console\_logging\_enabled**

若要停用主控台輸出記錄收集，請 `compute_console_logging_enabled` 將設定為 `false`。預設值為 `true`。

您可以隨時更新此參數，而不必停止運算叢集。

### **compute\_console\_logging\_max\_sample\_size**

`compute_console_logging_max_sample_size` 設定每次偵測到非預期終止時，從中 AWS ParallelCluster 收集主控台輸出的運算節點數目上限。如果此值小於 1，則會從所有終止的節點 AWS ParallelCluster 擷取主控台輸出。預設值為 1。

您可以隨時更新此參數，而不必停止運算叢集。

### **compute\_console\_wait\_time**

`compute_console_wait_time` 設定偵測到節點故障與從該節點收集控制台輸出之間的 AWS ParallelCluster 等待時間 (以秒為單位)。如果您確定 Amazon EC2 需要更多時間從終止的節點收集最終輸出，則可以增加等待時間。預設值為 300 秒 (5 分鐘)。

您可以隨時更新此參數，而不必停止運算叢集。

## 擷取AWS ParallelCluster UI 和AWS ParallelCluster執行階段記錄

了解如何擷取AWS ParallelCluster UI 和AWS ParallelCluster執行階段記錄以進行疑難排解。首先，請找到相關的AWS ParallelCluster UI 和AWS ParallelCluster堆棧名稱。使用堆疊名稱來尋找安裝記錄群組。若要完成，請匯出記錄檔。這些記錄檔專屬於AWS ParallelCluster執行階段。如需叢集記錄，請參閱[擷取和保留記錄](#)。

## 先決條件

- 已AWS CLI安裝。
- 您有憑據可以在AWS ParallelCluster UI 所在AWS 帳戶的AWS CLI命令上運行命令。
- 您可以在AWS ParallelCluster用戶界面上AWS 帳戶訪問亞馬遜 CloudWatch 控制台。

## 步驟 1：找到相關堆棧的堆棧名稱

在下列範例中，取代紅色反白文字作為您的實際值。

使用您安裝AWS ParallelCluster UI 的AWS 區域位置列出堆疊：

```
$ aws cloudformation list-stacks --region aws-region-id
```

請注意下列堆疊的堆疊名稱：

- 在您的帳戶上部署AWS ParallelCluster UI 的堆疊名稱。您在安裝AWS ParallelCluster UI 時輸入了此名稱；例如，pcluster-ui。
- 以您輸入的AWS ParallelCluster堆疊名稱為前置字元的堆疊；例如，pcluster-ui-ParallelClusterApi-ABCD1234EFGH。

## 步驟 2：找出記錄群組

列出AWS ParallelCluster UI 堆疊的日誌群組，如下列範例所示：

```
$ aws cloudformation describe-stack-resources \  
  --region aws-region-id \  
  --stack-name pcluster-ui \  
  --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' &&  
(LogicalResourceId == 'ApiGatewayAccessLog' || LogicalResourceId ==  
'ParallelClusterUILambdaLogGroup')].PhysicalResourceId" \  
  --output text
```

列出AWS ParallelCluster API 堆疊的日誌群組，如下列範例所示：

```
$ aws cloudformation describe-stack-resources \  
  --region aws-region-id \  
  --stack-name pcluster-ui-ParallelCluster-Api-ABCD1234EFGH \  
  --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' &&  
(LogicalResourceId == 'ApiGatewayAccessLog' || LogicalResourceId ==  
'ParallelClusterUILambdaLogGroup')].PhysicalResourceId" \  
  --output text
```

```
--query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' && LogicalResourceId == 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \  
--output text
```

請記下列日誌群組清單，以便在下列步驟中使用。

### 步驟 3：匯出日誌

請使用下列步驟來收集和匯出記錄檔：

1. 登入AWS Management Console，然後瀏覽至AWS ParallelCluster使用者介面所在AWS 帳戶的 [Amazon CloudWatch](#) 主控台。
2. 在導覽窗格中選擇 [記錄檔]、[記錄洞見]。
3. 選取上一個步驟中列出的所有日誌群組。
4. 選擇時間範圍，例如 12 小時。
5. 執行下列查詢：

```
$ fields @timestamp, @message  
| sort @timestamp desc  
| limit 10000
```

6. 選擇 [匯出結果]、[下載資料表 (JSON)]。

## 擷取和保留記錄

AWS ParallelCluster 為 HeadNode 和運算執行個體和儲存建立 Amazon EC2 指標。您可以在 CloudWatch 主控台「自訂儀表板」中檢視指標。AWS ParallelCluster 也會在 CloudWatch 記錄群組中建立叢集記錄串流。您可以在 CloudWatch 主控台「自訂儀表板」或「記錄」群組中檢視這些記錄。[監督叢集組態段落](#)說明如何修改叢集 CloudWatch日誌和儀表板。如需詳細資訊，請參閱 [與 Amazon Amazon Amazon CloudWatch 的整合](#) 及 [Amazon CloudWatch 儀表](#)。

記錄檔是疑難排解問題的有用資源。例如，如果您想要刪除失敗的叢集，首先建立叢集記錄的歸檔可能會很有用。依照中的步驟[封存記錄](#)建立歸檔。

### 主題

- [中無法使用叢集記錄 CloudWatch](#)
- [封存記錄](#)
- [保留的記錄](#)

- [終止的節點記錄](#)

## 中無法使用叢集記錄 CloudWatch

如果中無法使用叢集記錄 CloudWatch，請檢查以確定在將自訂記錄新增至配置時未覆寫記錄組態。  
AWS ParallelCluster CloudWatch

若要將自訂記錄新增至 CloudWatch 組態，請確定您附加至組態，而不是擷取並覆寫它。如需 fetch-config 和的詳細資訊 append-config，請參閱《CloudWatch 使用指南》中的「[多個 CloudWatch 代理程式組態檔](#)」。

若要還原記 AWS ParallelCluster CloudWatch 錄組態，您可以在 AWS ParallelCluster 節點內執行下列命令：

```
$ PLATFORM="$(ohai platform | jq -r ".[]")"
LOG_GROUP_NAME="$(cat /etc/chef/dna.json | jq -r ".cluster.log_group_name")"
SCHEDULER="$(cat /etc/chef/dna.json | jq -r ".cluster.scheduler")"
NODE_ROLE="$(cat /etc/chef/dna.json | jq -r ".cluster.node_type")"
CONFIG_DATA_PATH="/usr/local/etc/cloudwatch_agent_config.json"
/opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/python /usr/local/bin/write_cloudwatch_agent_json.py --platform $PLATFORM --config $CONFIG_DATA_PATH --log-group $LOG_GROUP_NAME --scheduler $SCHEDULER --node-role $NODE_ROLE
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json -s
```

## 封存記錄

您可以將日誌存檔在 Amazon S3 或本機檔案中 (視 --output-file 參數而定)。

### Note

將許可新增至 Amazon S3 儲存貯體政策以授予 CloudWatch 存取權。如需詳細資訊，請參閱 CloudWatch 日誌使用者指南中的在 [Amazon S3 儲存貯體上設定許可](#)。

```
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
  --bucket bucketname --bucket-prefix logs
{
  "url": "https://bucketname.s3.eu-west-1.amazonaws.com/export-log/mycluster-logs-202109071136.tar.gz?..."
```

```
}  
  
# use the --output-file parameter to save the logs locally  
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \  
  --bucket bucketname --bucket-prefix logs --output-file /tmp/archive.tar.gz  
{  
  "path": "/tmp/archive.tar.gz"  
}
```

除非在組態或`export-cluster-logs`命令的參數中明確指定，否則存檔包含過去 14 天來自頭節點和運算節點的 Amazon CloudWatch Logs 串流和 AWS CloudFormation 堆疊事件。命令完成所需的時間取決於叢集中的節點數目以及 CloudWatch 記錄檔中可用的記錄串流數目。如需有關可用記錄資料流的詳細資訊，請參閱[與 Amazon Amazon Amazon CloudWatch 的整合](#)。

## 保留的記錄

從版本 3.0.0 開始，刪除叢集時，預設 AWS ParallelCluster 會保留 CloudWatch 記錄檔。如果您想要刪除叢集並保留其記錄，請確定叢集配置Delete中[DeletionPolicy](#)未將[MonitoringLogsCloudWatch](#)設定為。否則，請將此欄位的值變更為Retain，然後執行`pcluster update-cluster`命令。然後，執行`pcluster delete-cluster --cluster-name <cluster_name>`以刪除叢集，但保留存放在 Amazon 中的日誌群組 CloudWatch。

## 終止的節點記錄

如果靜態計算節點意外終止，而且沒 CloudWatch 有記錄檔，請檢查是否 AWS ParallelCluster 已在記錄檔中的頭節點上記`/var/log/parallelcluster/compute_console_output`錄該計算節點的主控制台輸出。如需詳細資訊，請參閱[調試的關鍵日誌](#)。

如果記`/var/log/parallelcluster/compute_console_output`錄檔無法使用或不包含節點的輸出，請使用從失敗的節點擷取主控台輸出。AWS CLI 登入叢集頭節點，並`instance-id`從`/var/log/parallelcluster/slurm_resume.log`檔案中取得失敗的節點。

使用下列指令搭配下列命令來擷取主控台輸出`instance-id`：

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

如果動態計算節點在啟動後自行終止，且沒 CloudWatch 有記錄檔，請提交啟動叢集擴展動作的工作。等待執行個體失敗，並擷取執行個體主控台記錄。

登入叢集頭節點並`instance-id`從`/var/log/parallelcluster/slurm_resume.log`檔案取得計算節點。

使用下列命令擷取執行個體主控台記錄檔：

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

當運算節點記錄檔無法使用時，主控台輸出記錄檔可協助您偵錯運算節點失敗的根本原因。

## AWS CloudFormation 自訂資源

從 AWS ParallelCluster 版本 3.6.0 開始，您可以在 AWS CloudFormation 堆疊中使用 AWS ParallelCluster CloudFormation 自訂資源。自訂資源是 AWS ParallelCluster 託管堆疊。如此一來，您就可以用 CloudFormation 來設定和管理叢集。例如，您可以在 CloudFormation 堆疊中設定叢集外部資源，例如網路、共用儲存和安全群組基礎結構。此外，您可以使用 CloudFormation 基礎結構即程式碼管道來管理叢集。

執行下列動作，將 AWS ParallelCluster 自訂資源新增至 CloudFormation 範本：

1. 新增擁有和託管的自訂資源提供者堆疊 AWS ParallelCluster。
2. 將 CloudFormation 範本中的提供者堆疊參考為自訂資源。

自訂資源提供者堆疊會處理並回應要 CloudFormation 求。例如，當您部署 CloudFormation 堆疊時，也會設定並建立叢集。若要更新叢集，請更新 CloudFormation 堆疊。刪除堆疊時，請刪除叢集。如需有關 CloudFormation 自訂資源的詳細資訊，請參閱AWS CloudFormation 使用指南中的[自訂資源](#)。

### Warning

CloudFormation 不檢測自定義資源漂移。僅用 CloudFormation 於更新叢集配置和刪除叢集。您可以使用 [pcluster](#) CLI 或監視叢集的狀態或更新計算叢集，但不能使用它們來更新叢集配置或刪除叢集。[AWS ParallelCluster 界面](#)

### Note

我們建議您在堆疊中加入[終止保護](#)，以避免意外移除。

## 提供者堆疊由 AWS ParallelCluster

自訂資源提供者堆疊的格式如下列 CloudFormation 範本程式碼片段所示：

```
PclusterClusterProvider:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      CustomLambdaRole: # (Optional) RoleARN to override default
      AdditionalIamPolicies: # (Optional) comma-separated list of IAM policies to add
    TemplateURL: !Sub
      - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.${AWS::URLSuffix}/
parallelcluster/${Version}/templates/custom_resource/cluster.yaml
      - { Version: 3.7.0 }
```

## 屬性：

### 參數：

#### CustomLambdaRole (可選)：

具有執行權限的自訂角色 AWS Lambda，可建立和管理叢集。根據預設，角色會使用[AWS ParallelCluster 文件](#)中預設定義的相同原則。

#### AdditionalIamPolicies (可選)：

以逗號分隔的其他 IAM 政策清單，其中包含要新增至 Lambda 使用之角色的 Amazon 資源名稱 (ARN)。這僅在 CustomLambdaRole 未指定並且可以保持空白時才使用。

如果您需要額外的標頭節點、運算節點或存取 Amazon S3 儲存貯體的 policy，請將它們新增到 CustomLambdaRole 或 AdditionalIamPolicy 屬性。

如需有關預設原則的詳細資訊，請參閱[AWS Identity and Access Management 權限 AWS ParallelCluster](#)。

#### TemplateURL (必填)：

自 AWS ParallelCluster 訂資源檔案 URL。

## 輸出：

### ServiceToken:

可用作自訂資源 ServiceToken 內容的值。自訂資源會 ServiceToken 指定 AWS CloudFormation 傳送請求的位置。這是包含在 AWS CloudFormation 範本中之叢集資源的必要輸入。



**LogGroupArn:**

基礎資源記錄到的 ARN。CloudWatch LogGroup

**LambdaLayerArn:**

用於執行 AWS ParallelCluster 作業的 Lambda 層的 ARN。

## 叢集資源

CloudFormation 叢集資源的格式如下列 CloudFormation 範本程式碼片段所示：

```
PclusterCluster:
  Type: Custom::PclusterCluster
  Properties:
    ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
    ClusterName: !Sub 'c-${AWS::StackName}' # Must be different from StackName
    ClusterConfiguration:
      # Your Cluster Configuration
```

**屬性：****ServiceToken:**

提 AWS ParallelCluster 供者堆疊ServiceToken輸出。

**ClusterName:**

要建立和管理的叢集名稱。名稱不得與 CloudFormation 堆疊的名稱相符。建立叢集之後，就無法變更名稱。

**ClusterConfiguration:**

叢集配置 YAML 檔案，如中[叢集配置檔](#)所述。不過，您可以使用一般的 CloudFormation 建構，例如[內建函式](#)。

**DeletionPolicy:**

定義刪除根堆疊時是否刪除叢集。預設值為 Delete。

**保留：**

如果刪除自訂資源，請保留叢集。

**Note**

若要維持保留的叢集運作，叢集相依資源 (例如儲存區和網路) 必須將刪除原則設定為保留。

刪除：

刪除自訂資源時，請刪除叢集。

**Fn::GetAtt**返回值：

Fn::GetAtt內在函數返回一個類型的指定屬性的值。如需有關使用Fn::GetAtt intrinsic函數的詳細資訊，請參閱 [Fn:: GetAtt](#)。

ClusterProperties:

作業中的 [pcluster describe-cluster](#) 值。

驗證訊息：

字串，其中包含上次建立或更新作業期間發生的所有驗證訊息。

logGroupName:

用於記錄 Lambda 叢集作業的日誌群組名稱。記錄事件會保留 90 天，並在刪除叢集後保留記錄群組。

範例:Fn::GetAtt:

```
# Provide the public IP address of the head node as an output of a stack
Outputs:
  HeadNodeIp:
    Description: The public IP address of the head node
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]
```

範例：具有 AWS ParallelCluster 自訂資源的簡單完整 CloudFormation 範本：

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
  HeadNodeSubnet:
```

```
Description: Subnet where the HeadNode will run
Type: AWS::EC2::Subnet::Id

ComputeSubnet:
  Description: Subnet where the Compute Nodes will run
  Type: AWS::EC2::Subnet::Id

KeyName:
  Description: KeyPair to login to the head node
  Type: AWS::EC2::KeyPair::KeyName

Resources:
  PclusterClusterProvider:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
        - { Version: 3.7.0 }

  PclusterCluster:
    Type: Custom::PclusterCluster
    Properties:
      ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
      ClusterName: !Sub 'c-${AWS::StackName}'
      ClusterConfiguration:
        Image:
          Os: alinux2
        HeadNode:
          InstanceType: t2.medium
          Networking:
            SubnetId: !Ref HeadNodeSubnet
          Ssh:
            KeyName: !Ref KeyName
        Scheduling:
          Scheduler: slurm
          SlurmQueues:
            - Name: queue0
              ComputeResources:
                - Name: queue0-cr0
                  InstanceType: t2.micro
              Networking:
                SubnetIds:
                  - !Ref ComputeSubnet
```

**Outputs:****HeadNodeIp:**

Description: The Public IP address of the HeadNode

Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]

**ValidationMessages:**

Description: Any warnings from cluster create or update operations.

Value: !GetAtt PclusterCluster.validationMessages

若要進一步瞭解如何使用 CloudFormation AWS ParallelCluster 自訂資源，請參閱[建立叢集 AWS CloudFormation](#)。

## 叢集作業

將叢集自訂資源新增至 CloudFormation 堆疊時，CloudFormation 可以執行下列叢集作業：

- CloudFormation 當叢集部署包含 AWS ParallelCluster 自訂資源的堆疊時，會在新的個別堆疊中建立叢集。
- 如果您更新堆疊中定義的叢集配置，則會根據組態更新原則 CloudFormation 更新叢集。在更新叢集之前，AWS ParallelCluster 自訂資源提供者不會停止運算叢集。建議您使用此[QueueUpdateStrategy](#)設定進行叢集更新。如此一來，您就可以避免在使用 AWS ParallelCluster 自訂資源時，在更新前後進行明確的pcluster update-compute-fleet呼叫。
- 如果刪除堆疊，則會刪除叢集。

## 疑難排解包含 AWS ParallelCluster 自訂資源的堆疊

使用 AWS ParallelCluster 自訂資源，從新的個別堆疊 CloudFormation 部署叢集。您可以執行下列步驟來監視叢集建立：

1. 導覽至，AWS Management Console 然後 CloudFormation 在導覽窗格中選擇「堆疊」。
2. 使用您為叢集名稱定義的名稱選擇堆疊。
3. 如果堆疊狀態為ROLLBACK\_COMPLETE，則表示叢集建立期間發生錯誤。
4. 選擇堆疊詳細資料，然後選擇事件索引標籤。
5. 針對您為叢集名稱定義的名稱，搜尋邏輯 ID 上的事件。它有一Status reason個問題的原因。
6. 您也可以選擇「堆疊」下拉式選單，然後選擇「已刪除」來查看已刪除堆疊的清單。選取具有叢集名稱的堆疊，並檢視事件以取得更多詳細資訊。

- 若要檢視管理叢集之自訂資源提供者的輸出，請選取包含說明「AWS ParallelCluster 叢集自訂資源」的堆疊。選擇資源選項卡，找到具有邏輯 ID 的資源 `PclusterCfnFunctionLogGroup`，然後跟隨給定的鏈接。檢視顯示 Lambda 除錯輸出的日誌串流。
- 若要疑難排解叢集，請參閱 [AWS ParallelCluster 疑難排](#)。

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) 是一種具有 OS-bypass 功能的網路裝置，可與相同子網路上的其他執行個體進行低延遲網路通訊。EFA 是使用 Libfabric 公開，並且可以使用簡訊傳遞界面 (MPI) 供應用程式使用。

若要搭配使用 EFA AWS ParallelCluster 和 Slurm 排程器，請將 [SlurmQueues//ComputeResourcesEfa](#) 設定 `Enabled` 為 `true`。

若要檢視支援 EFA 的 Amazon 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](#)。

建議您在置放群組中執行啟用 EFA 的執行個體。如此一來，執行個體就會啟動到單一可用區域中的低延遲群組。如需如何使用配置置放群組的詳細資訊 AWS ParallelCluster，請參閱 [SlurmQueuesNetworking//PlacementGroup](#)。

如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [彈性網狀架構適配器](#)，以及使用 [彈性網狀架構介面卡](#) 和 [AWS 開放原始碼部落格 AWS ParallelCluster 中擴展 HPC 工作負載](#)。

### Note

不同的可用區域不支援 Elastic Fabric Adapter (EFA)。如需詳細資訊，請參閱 [排程 SlurmQueues//網路/SubnetIds](#)。

### Note

根據預設，Ubuntu 分發會啟用 ptrace (程序追蹤) 保護。ptrace 保護被禁用，以便利布正常工作。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 [停用 ptrace 保護](#)。

## 啟用英特爾 MPI

英特爾MPI是可用的 AWS ParallelCluster AMIs。

### Note

若要使用 IntelMPI，您必須瞭解並接受 [Intel 簡化軟體授權](#) 的條款。

依預設，「開啟」MPI 會放置在路徑上。若要啟用 Intel MPI 而不是「開啟」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                modules
libfabric-aws/1.16.0~amzn3.0  null
module-git         openmpi/4.1.4
module-info        use.own

-----/opt/intel/mpi/2021.6.0/modulefiles
-----
intelmpi
```

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 intelmpi
```

要查看哪些模組已載入，請執行 `module list`。

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

若要確認 Intel MPI 是否已啟用，請執行 `mpirun --version`。

```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2021.6 Build 20220227 (id: 28877f3f32)
Copyright 2003-2022, Intel Corporation.
```

載入 Intel MPI 模組之後，會變更多個路徑以使用 Intel MPI 工具。若要執行由 Intel MPI 工具編譯的程式碼，請先載入 Intel MPI 模組。

#### Note

Intel 與以重力為 AWS 基礎的執行個體MPI不相容。

#### Note

在 2.5.0 AWS ParallelCluster 版之前，Intel 尚MPI未 AWS ParallelCluster AMIs在中國 (北京) 和中國 (寧夏) 地區提供。

## AWS ParallelCluster API

什麼是 AWS ParallelCluster API ?

AWS ParallelCluster API 是一種無伺服器應用程式，一旦部署到您的應用程式 AWS 帳戶，就 AWS ParallelCluster 能透過 API 以程式設計方式存取功能。

AWS ParallelCluster API 是以獨立[AWS CloudFormation](#)範本的形式散佈，其中包含 [Amazon API Gateway](#) 端點、公開 AWS ParallelCluster [AWS Lambda](#)功能以及負責處理叫用功能的函數。

下圖顯示了 AWS ParallelCluster API 基礎結構的高級架構圖。

## AWS ParallelCluster API 文件

您可以從下列網站下載描述 AWS ParallelCluster API 的 OpenAPI 規格檔案：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

從 [OpenAPI 規範檔案開始](#)，您可以使用許多可用工具之一，例如 [Swagger UI](#) 或 [Redoc](#) 來產生 [AWS ParallelCluster API 的文件](#)。

如何部署 AWS ParallelCluster API

若要部署 AWS ParallelCluster API，您必須是 AWS 帳戶。

用於部署 API 的範本可從下列網址取得：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

API 需要部署到 AWS 區域 哪裡，<VERSION>是 AWS ParallelCluster 版本（例如 3.7.0）。<REGION>

AWS Lambda 透過使用 Lambda 層介面與[AWS ParallelClusterPython 函式庫](#)。

#### Warning

中任何具有權 AWS 帳戶限存取 AWS Lambda 或 Amazon API Gateway 服務的使用者都會自動繼承管理 AWS ParallelCluster API 資源的許可。

## 使用部署 AWS CLI

如果尚未設定要與 CLI 搭配使用的 AWS 認證。

```
$ aws configure
```

執行下列命令以部署 API：

```
$ REGION=<region>  
$ API_STACK_NAME=<stack-name> # This can be any name  
$ VERSION=3.7.0  
$ aws cloudformation create-stack \  
  --region ${REGION} \  
  --stack-name ${API_STACK_NAME} \  
  --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/  
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \  
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND  
$ aws cloudformation wait stack-create-complete --stack-name ${API_STACK_NAME} --region  
${REGION}
```

### 自訂您的部署

您可以使用範本公開的 AWS CloudFormation 參數來自訂 API 部署。若要在透過 CLI 部署時設定參數值，可以使用下列選項：`--parameters ParameterKey=KeyName,ParameterValue=Value`



以下參數可選：

- 區域-使用Region參數來指定 API 是否能夠控制所有 AWS 區域 (預設) 或單一資源 AWS 區域。將此值設定為要部署的 API，以限制存取。AWS 區域
- ParallelClusterFunctionRole-這會覆寫指派給實作功能的 AWS Lambda 函數的 IAM 角色 AWS ParallelCluster 色。此參數可接受 IAM 角色的 ARN。此類角色必須設定 AWS Lambda 為具有 IAM 主體。
- CustomDomainNameCustomDomainCertificate、CustomDomainHostedZoneId-使用這些參數為 Amazon API Gateway 端點設定自訂網域。CustomDomainName是要使用的網域名稱，CustomDomainCertificate是此網域名稱的 AWS 受管憑證的 ARN，CustomDomainHostedZoneId是您要在其中建立記錄的 [Amazon Route 53](#) 託管區域的識別碼。

**Warning**

您可以設定自訂網域設定，為 API 強制執行傳輸層安全性 (TLS) 的最低版本。如需詳細資訊，請參閱在 [API Gateway 中為自訂網域選擇最低 TLS 版本](#)。

- EnableIamAdminAccess-默認情況下，AWS Lambda 函數處理 AWS ParallelCluster API 操作配置為防止任何特權 IAM 訪問 ( EnableIamAdminAccess=false ) 的 IAM 角色。這會讓 API 無法處理需要建立 IAM 角色或政策的作業。因此，只有當 IAM 角色作為資源配置的一部分提供作為輸入時，才能成功建立叢集或自訂映像。

設定EnableIamAdminAccess為 AWS ParallelCluster API 時，true將獲得許可，以管理部署叢集或產生自訂 AMI 所需的 IAM 角色建立。

**Warning**

將此項設定為 true 會授予 AWS Lambda 函數處理 AWS ParallelCluster 作業的 IAM 管理員權限。

如需啟用此模式時可解除鎖定之功能的其他詳細資訊，請參閱。 [AWS ParallelCluster 管理 IAM 資源的使用者範例政策](#)

- PermissionsBoundaryPolicy-此選用參數可接受現有的 IAM 政策 ARN，該 ARN 將設定為 PC API 基礎結構建立的所有 IAM 角色的許可界限，並作為管理 IAM 許可的條件，以便 PC API 只能建立具有此政策的角色。

如需[PermissionsBoundary 模式](#)有關此模式所施加之限制的其他詳細資訊，請參閱。

- `CreateApiUserRole`-依預設，AWS ParallelCluster API 的部署包括建立 IAM 角色，該角色設定為唯一獲得授權呼叫 API 的角色。Amazon API Gateway 端點設定了以資源為基礎的政策，以僅將叫用權限授與建立的使用者。若要變更此設定，請設定 API 存取權，`CreateApiUserRole=false`然後將其授與給選取的 IAM 使用者。如需詳細資訊，請參閱 Amazon API 閘道開發人員指南中的控制叫用 API [的存取](#)。

#### Warning

如果對 API 端點的`CreateApiUserRole=true`存取未受到 Amazon API Gateway 資源政策的限制，所有具有不受限制`execute-api:Invoke`權限的 IAM 角色都可以存取 AWS ParallelCluster 功能。如需詳細資訊，請參閱《[API Gateway 開發人員指南](#)》中的[使用 API Gateway 資源政策控制](#) API 的存取。

#### Warning

`ParallelClusterApiUserRole`具有叫用所有 AWS ParallelCluster API 作業的權限。若要限制 API 資源子集的存取權限，請參閱《[API Gateway 開發人員指南](#)》中的[控管人員可以使用 IAM 政策呼叫 API Gateway API 方法](#)。

- `IAM RoleAndPolicyPrefix`-此選用參數最多可接受 10 個字元的字串，該字串將用作 IAM 角色和建立為 PC API 基礎結構一部分的政策的前綴。

## 正在更新 API

升級至較新的 AWS ParallelCluster 版本

選項 1：通過刪除相應的 AWS CloudFormation 堆棧並部署新的 API 來刪除現有的 API，如上所示。

選項 2：通過運行以下命令更新現有的 API：

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.7.0
$ aws cloudformation update-stack \
  --region ${REGION} \
```

```
--stack-name ${API_STACK_NAME} \  
--template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/  
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \  
--capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND  
$ aws cloudformation wait stack-update-complete --stack-name ${API_STACK_NAME} --region  
${REGION}
```

## 調用 API AWS ParallelCluster

AWS ParallelCluster Amazon API Gateway 端點設定為[AWS\\_IAM授權類型](#)，並要求所有請求都必須使用有效的 IAM 登入資料簽署 ([API 參考：發出 http 請求](#))。

使用預設設定部署時，API 叫用許可僅授與使用 API 建立的預設 IAM 使用者。

若要擷取預設 IAM 使用者的 ARN，請執行：

```
$ REGION=<region>  
$ API_STACK_NAME=<stack-name>  
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}  
--query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiUserRole'].OutputValue" --  
output text
```

若要取得預設 IAM 使用者的臨時登入資料，請執行 [STS AssumeRole](#) 命令。

您可以執行下列命令來擷取 AWS ParallelCluster API 端點：

```
$ REGION=<region>  
$ API_STACK_NAME=<stack-name>  
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}  
--query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiInvokeUrl'].OutputValue" --  
output text
```

該 AWS ParallelCluster API 可以由任何符合 OpenAPI 規範的 HTTP 客戶端調用，可以在這裡找到：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

請求需要[按照](#)此處的記錄進行 SIGv4 簽名。

目前，我們不提供任何官方 API 客戶端實現。[不過，API 用戶端可以透過使用 OpenAPI 產生器輕鬆地從 OpenAPI 模型產生。](#)一旦客戶端生成，如果沒有提供開箱即用，則需要添加 SIGv4 簽名。

您可以在[AWS ParallelCluster 儲存庫](#)中找到 Python API 用戶端的參考實作。若要進一步瞭解如何使用 Python API 用戶端，請參閱[應用 AWS ParallelCluster 程式介面教學課程](#)。

若要實作更進階的存取控制機制 (例如 Amazon Cognito 或 Lambda 授權器)，或要進一步保護使用 AWS WAF 或 API 金鑰的 API，請遵循 [Amazon API Gateway](#) 文件。

#### Warning

獲授權可叫用 AWS ParallelCluster API 的 IAM 使用者可間接控制 AWS ParallelCluster 中管理的所有 AWS 資源 AWS 帳戶。這包括建立由於使用者 IAM 政策的限制，使用者無法直接控制的 AWS 資源。例如，建立 AWS ParallelCluster 叢集時，視其組態而定，可能包括部署 Amazon EC2 執行個體、Amazon Route 53、Amazon Elastic File System 檔案系統、Amazon FSx 檔案系統、IAM 角色，以及其他 AWS 服務 使用者 AWS ParallelCluster 可能無法直接控制的資源。

#### Warning

使用配置中 `AdditionalIamPolicies` 指定的建立叢集時，其他原則必須符合下列其中一種模式：

```
- !Sub arn:${AWS::Partition}:iam:${AWS::AccountId}:policy/parallelcluster*
- !Sub arn:${AWS::Partition}:iam:${AWS::AccountId}:policy/parallelcluster/*
- !Sub arn:${AWS::Partition}:iam:aws:policy/CloudWatchAgentServerPolicy
- !Sub arn:${AWS::Partition}:iam:aws:policy/AmazonSSMManagedInstanceCore
- !Sub arn:${AWS::Partition}:iam:aws:policy/AWSBatchFullAccess
- !Sub arn:${AWS::Partition}:iam:aws:policy/AmazonS3ReadOnlyAccess
- !Sub arn:${AWS::Partition}:iam:aws:policy/service-role/AWSBatchServiceRole
- !Sub arn:${AWS::Partition}:iam:aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role
- !Sub arn:${AWS::Partition}:iam:aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
- !Sub arn:${AWS::Partition}:iam:aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole
- !Sub arn:${AWS::Partition}:iam:aws:policy/EC2InstanceProfileForImageBuilder
- !Sub arn:${AWS::Partition}:iam:aws:policy/service-role/
AWSLambdaBasicExecutionRole
```

如果您需要其他政策，可以執行下列其中一項操作：

- 編輯DefaultParallelClusterIamAdminPolicy中的：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

在ArnLike/iam:PolicyARN區段中新增原則。

- 省略在組態檔AdditionalIamPolicies中指定原則，並手動將原則新增至叢集內建立的 AWS ParallelCluster 執行個體角色。

## 訪問 API 日誌和指標

API 日誌會發佈到 Amazon CloudWatch，保留期限為 30 天。若要擷取與 API 部署相關聯的 LogGroup 名稱，請執行下列命令：

```
$ REGION=<region>  
$ API_STACK_NAME=<stack-name>  
$ aws cloudformation describe-stacks --region ${REGION} --  
stack-name ${API_STACK_NAME} --query "Stacks[0].Outputs[?  
OutputKey=='ParallelClusterLambdaLogGroup'].OutputValue" --output text
```

Lambda 指標、日誌和[AWS X-Ray](#)追蹤日誌也可以透過 Lambda 主控台存取。若要擷取與 API 部署相關聯之 Lambda 函數的 ARN，請執行下列命令：

```
$ REGION=<region>  
$ API_STACK_NAME=<stack-name>  
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}  
--query "Stacks[0].Outputs[?OutputKey=='ParallelClusterLambdaArn'].OutputValue" --  
output text
```

## AWS ParallelCluster 對於地形

從 AWS ParallelCluster 3.8.0 開始，您可以使用 [Terraform](#) 部署叢集和自訂映像檔。若要開始使用此功能，請參閱 [Terraform 登錄 AWS ParallelCluster 中的地形表單提供者](#)。

**Note**

您必須在帳戶中部署 [ParallelCluster API](#) 才能使用提供者。

請使用下列圖表判斷提供者與 AWS ParallelCluster 版本之間的相容性：

| 提供者版本 | AWS ParallelCluster 版本 |
|-------|------------------------|
| 1.0.0 | 3.8.0+                 |

請參閱[如何使用提供者的範例](#)。

為了獲得更流暢的體驗，請使用官方 [Terraform 模塊 AWS ParallelCluster](#) 從 Terraform 註冊表。此模組可讓您部署：

1. ParallelCluster API
2. ParallelCluster 使用 YAML 設定檔和 HCL 定義的叢集
3. ParallelCluster 叢集所需的網路基礎結構

請參閱[如何使用模組的範例](#)。

## 通過 NICE DCV Connect 到頭節點

NICE DCV 是一種遠端視覺化技術，可讓使用者安全地連接到遠端高效能伺服器上託管的圖形密集型 3D 應用程式。如需詳細資訊，請參閱 [NICE DCV](#)。

NICE DCV 軟體會自動安裝在頭節點上，並且可以使用 [HeadNode](#) 組態中的 [Dcv](#) 部分來啟用。

```
HeadNode:
  Dcv:
    Enabled: true
```

這樣，在頭節點 `/home/<DEFAULT_AMI_USER>` 中將 AWS ParallelCluster 設置為 [DCV 服務器存儲文件夾](#)。如需有關 NICE DCV 組態參數的詳細資訊，請參閱 [HeadNode/Dcv](#)。若要連線至 NICE DCV 工作階段，請使用指 [pcluster dcv-connect](#) 令。

## NICE DCV DCV HTTPS 證書

NICE DCV 會自動產生自我簽署的憑證，以保護 NICE DCV 用戶端和 NICE DCV 伺服器之間的流量。

若要將預設的自我簽署 NICE DCV 憑證取代為另一個憑證，請先連線至標頭節點。然後，在執行 `pcluster dcv-connect` 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《NICE DCV 管理員指南》中的[變更 TLS 憑證](#)。

## NICE DCV 發牌

在 Amazon EC2 執行個體上執行時，NICE DCV 伺服器不需要授權伺服器。不過，NICE DCV 伺服器必須定期連線到 Amazon S3 儲存貯體，以判斷是否有效的授權可用。

AWS ParallelCluster 自動將必要的許可新增至標頭節點 IAM 政策。使用自訂 IAM 執行個體政策時，請使用 [NICE DCV 管理員指南中 Amazon EC2 上的 NICE DCV](#) 中所述的許可。

如需疑難排解秘訣，請參閱 [疑難排解 NICE DCV 中的問題](#)。

## 使用 `pcluster update-cluster`

在 AWS ParallelCluster 3.x 中，`pcluster update-cluster` 分析用於建立目前叢集的設定，以及配置檔案中的設定是否有問題。如果發現任何問題，就會回報問題，並顯示修正問題所採取的步驟。例如，如果計算已變更，`InstanceType` 則必須先停止運算叢集，才能繼續進行更新。發現此問題時，就會報告此問題。如果未發現任何封鎖問題，則會啟動更新程序並報告變更。

您可以使用在 `pcluster update-cluster --dryrun` option 執行變更之前查看變更。如需詳細資訊，請參閱 [pcluster update-cluster 範例](#)。

如需疑難排解指引，請參閱 [AWS ParallelCluster 疑難排](#)。

## 更新原則：定義

更新原則：您可以在更新期間變更此設定。

變更此設定後，可以使用更新叢集 `pcluster update-cluster`。

更新原則：如果變更此設定，則不允許更新。

變更此設定後，就無法更新叢集。您必須還原原始叢集的設定，並使用更新的設定建立新叢集。您可以稍後刪除原始叢集。若要建立新叢集，請使用 `pcluster create-cluster`。若要刪除原始叢集，請使用 `pcluster delete-cluster`。

更新原則：更新期間不會分析此設定。

您可以變更這些設定，並使用更新叢集[pcluster update-cluster](#)。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

當運算叢集存在時，無法變更這些設定。必須還原變更，或者必須停止運算叢集 (使用[pcluster update-compute-fleet](#))。計算叢集停止後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟動變更。例如，如果您使用Slurm排程器的 [SlurmQueues/ComputeResources/-Name/MinCount](#) > 0，則會啟動計算叢集。

更新原則：必須停止計算叢集和登入節點，才能變更此設定以進行更新。

當運算叢集存在或登入節點正在使用中時，無法變更這些設定。必須還原變更，或者必須停止運算叢集和登入節點 (可以使用停止運算叢集[pcluster update-compute-fleet](#))。停止運算叢集和登入節點之後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟動變更。

更新原則：更新期間無法降低此設定。


您可以變更這些設定，但無法減少。如果必須減少這些設定，您必須還原原始叢集的設定，並使用更新的設定建立新叢集。您可以稍後刪除原始叢集。若要建立新叢集，請使用[pcluster create-cluster](#)。若要刪除原始叢集，請使用[pcluster delete-cluster](#)。

更新原則：如果變更此設定，則不允許更新。如果強制更新，則會忽略新值，並使用舊值。

變更此設定後，就無法更新叢集。您必須還原原始叢集的設定，並使用更新的設定建立新叢集。您可以稍後刪除原始叢集。若要建立新叢集，請使用[pcluster create-cluster](#)。若要刪除原始叢集，請使用[pcluster delete-cluster](#)。

更新原則：必須停止運算叢集，或者[QueueUpdateStrategy](#)必須設定，才能變更此設定以進行更新。

您可以變更這些設定。必須停止 (使用[pcluster update-compute-fleet](#)) 或設[QueueUpdateStrategy](#)定運算叢集。停止或[QueueUpdateStrategy](#)設定運算叢集之後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟動變更。

 Note

此更新政策從 3.2.0 AWS ParallelCluster 版開始受到支援。



更新原則：對於此清單值設定，可以在更新期間新增新值，或者在移除現有值時必須停止運算叢集。

您可以在更新期間新增這些設定的新值。將新值添加到列表後，可以使用 ( [pcluster update-cluster](#) ) 更新集群。

若要從清單中移除現有值，必須停止運算叢集 (使用 [pcluster update-compute-fleet](#))。

例如，如果您使用Slurm排程器並將新的執行個體類型新增至 [Instance/InstanceType](#)，則可以更新叢集而不必停止運算叢集。[若要從執行個體/移除現有的執行個體類型InstanceType](#)，必須先停止運算叢集 (使用 [pcluster update-compute-fleet](#))。

**Note**

此更新政策從 3.2.0 AWS ParallelCluster 版開始受到支援。

更新原則：減少佇列的大小需要停止運算叢集，或者 [QueueUpdateStrategy](#) 必須設定為「終止」，才能變更此設定以進行更新。

您可以變更這些設定，但是如果變更會減少佇列的大小，則必須停止運算叢集 (使用 [pcluster update-compute-fleet](#))，或者 [QueueUpdateStrategy](#) 必須設定為「終止」。在計算叢集停止或設定 [QueueUpdateStrategy](#) 為「終止」之後，您可以更新叢集 ([pcluster 更新叢集](#)) 以啟動變更。

調整叢集容量大小時設定的「終止」設定，只會從節點清單的背面終止節點，而且不會影響同一個分割區的所有其他節點。

例如，如果叢集初始容量為 `MinCount = 5` 和 `MaxCount = 10`，則節點為 `st-[1-5]`；`dy-[1-5]`。將叢集大小調整為 `MinCount = 3` 和 `MaxCount = 5`，新的叢集容量將由節點組成 `st-[1-3]`；`dy-[1-2]`，這些節點在更新期間將不會觸及。只 `st-[4-5]`；`dy-[3-5]` 有節點會在更新期間終止。

支援下列變更，且不需要停止運算叢集，也不需要將其 [QueueUpdateStrategy](#) 設定為「終止」：

- 一個新的 [SlurmQueue](#) 被添加
- 一個新的 [ComputeResource](#) 被添加
- [MaxCount](#) 增加
- [MinCount](#) 增加並 [MaxCount](#) 增加至少相同數量

注意：從 AWS ParallelCluster 版本 3.9.0 開始支援此更新政策。

更新原則：對於此清單值設定，必須停止運算叢集，或者 [QueueUpdateStrategy](#) 必須設定為新增值；移除現有值時必須停止運算叢集。

您可以在更新期間新增這些設定的新值。必須停止 (使用 [pcluster update-compute-fleet](#)) 或設 [QueueUpdateStrategy](#) 定運算叢集。停止或 [QueueUpdateStrategy](#) 設定運算叢集之後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟動變更。

若要從清單中移除現有值，必須停止運算叢集 (使用 [pcluster update-compute-fleet](#))。

#### Note

此更新政策從 3.3.0 AWS ParallelCluster 版開始受到支援。

更新原則：必須停止所有運算節點，才能刪除受管理的置放群組。必須停止或設定計算叢集，[QueueUpdateStrategy](#) 才能變更此設定以進行更新。

必須停止運算叢集 (使用 [pcluster update-compute-fleet](#))，才能移除受管理的置放群組。如果您在停止計算叢集之前執行叢集更新以移除受管理的置放群組，則會傳回無效的設定訊息，而且不會繼續進行更新。停止運算叢集可確保沒有執行個體正在執行。

## pcluster update-cluster 範例

您可以變更這些設定，但是如果變更會減少佇列的大小，則必須停止運算叢集 (使用 [pcluster update-compute-fleet](#))，或者 [QueueUpdateStrategy](#) 必須設定為「終止」。在計算叢集停止或設定 [QueueUpdateStrategy](#) 為「終止」之後，您可以更新叢集 ([pcluster 更新叢集](#)) 以啟動變更。

- 這個範例會示範包含一些允許變更的更新，而且會直接啟動更新。

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
  "cluster": {
    "clusterName": cluster_name,
    "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
    "cloudformationStackArn": stack_arn,
    "region": "us-east-1",
    "version": "3.7.0",
    "clusterStatus": "UPDATE_IN_PROGRESS"
  },
}
```

```

"changeSet": [
  {
    "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
    "requestedValue": [
      "sg-0cd61884c4ad11234"
    ],
    "currentValue": [
      "sg-0cd61884c4ad16341"
    ]
  }
]
}

```

- 這個範例會示範包含一些允許變更的 dryrun 更新。Dryrun 在不啟動更新的情況下報告變更集非常有用。

```

$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/parallelcluster/test_cluster --region us-east-1 --dryrun true
{
  "message": "Request would have succeeded, but DryRun flag is set.",
  "changeSet": [
    {
      "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
      "requestedValue": [
        "sg-0cd61884c4ad11234"
      ],
      "currentValue": [
        "sg-0cd61884c4ad16341"
      ]
    }
  ]
}

```

- 這個例子演示了一些阻止更新的更新。

```

$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/parallelcluster/test_cluster --region us-east-1
{
  "message": "Update failure",
  "updateValidationErrors": [
    {
      "parameter": "HeadNode.Ssh.KeyName",
      "requestedValue": "mykey_2",

```

```

    "message": "Update actions are not currently supported for the 'KeyName'
parameter. Restore 'KeyName' value to 'jenkinsjun'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
    "currentValue": "mykey_1"
  },
  {
    "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
    "requestedValue": "c4.xlarge",
    "message": "All compute nodes must be stopped. Stop the compute fleet with the
pcluster update-compute-fleet command",
    "currentValue": "t2.micro"
  },
  {
    "parameter": "SharedStorage[ebs1].MountDir",
    "requestedValue": "/my/very/very/long/shared_dir",
    "message": "Update actions are not currently supported for the 'MountDir'
parameter. Restore 'MountDir' value to '/shared'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
    "currentValue": "/shared"
  }
],
"changeSet": [
  {
    "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
    "requestedValue": [
      "sg-0cd61884c4ad11234"
    ],
    "currentValue": [
      "sg-0cd61884c4ad16341"
    ]
  },
  {
    "parameter": "HeadNode.Ssh.KeyName",
    "requestedValue": "mykey_2",
    "currentValue": "mykey_1"
  },
  {
    "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
    "requestedValue": "c4.xlarge",
    "currentValue": "t2.micro"
  },
  {

```

```
    "parameter": "SharedStorage[ebs1].MountDir",
    "requestedValue": "/my/very/very/long/shared_dir",
    "currentValue": "/shared"
  }
]
}
```

## AWS ParallelCluster AMI 定制

在某些情況下，建置自訂 AMI AWS ParallelCluster 是必要的。本節涵蓋構建自定義 AWS ParallelCluster AMI 時應考慮的事項。

您可以使用下列其中一種方法建立自訂 AWS ParallelCluster AMI：

1. 建立組建映像設定檔，然後使用 `pcluster` CLI 透過 EC2 Image Builder 建置映像。此過程是自動化的，可重複的，並支持監控。若要取得更多資訊，請參閱 [pcluster](#) 映像指令。
2. 從 AWS ParallelCluster AMI 建立執行個體，然後登入並進行手動修改。最後，使用 Amazon EC2 從修改後的執行個體建立新 AMI。這個過程需要更少的時間。但是，它不是自動化或可重複的，並且不支援使用 `pcluster` CLI 映像監視命令。

如需這些方法的詳細資訊，請參閱 [構建自定義 AWS ParallelCluster AMI](#)。

## AWS ParallelCluster AMI 自訂考量

無論您如何建立自訂映像檔，我們都建議您執行初步驗證測試，並包含規定以監控所建立映像的狀態。

若要使用建置自訂 AMI `pcluster`，請使用 [EC2 Image Builder](#) 用來建立自訂映像檔的 `Build` 和 `Image` 區段來建立自訂映像檔。此 `Build` 區段會指定建置映像所需的影像產生器。這包括 `ParentImage`（基本映像）和 `Components`。[Image Builder 元件](#) 會定義一系列步驟，這些步驟是在建立映像檔之前自訂執行個體，或測試由建立的映像檔啟動的執行個體所需的步驟。如需 AWS ParallelCluster 元件範例，請參閱 [自訂 AMI](#)。該 `Image` 部分指定圖像屬性。

當從 `pcluster` 呼叫 `build-image` 以建立自訂映像檔時，Image Builder 會使用組建映像檔組態與說 AWS ParallelCluster 明書來啟動您 AWS ParallelCluster 的 `ParentImage` Image Builder 會下載元件、執行建置和驗證階段、建立 AMI、從 AMI 啟動執行個體，以及執行測試。程序完成後，Image Builder 會產生新的影像或停止訊息。

## 執行自訂元件驗證測試

在組態中包含 Image Builder 元件之前，請使用下列其中一種方法對其進行測試和驗證。由於 Image Builder 程序最多可能需要 1 小時，因此建議您事先測試元件。這可以為您節省大量時間。

### 腳本大小寫

在建置映像處理序之外的執行中執行個體中測試指令碼，並確認指令碼是否以結束代碼 0 結束。

### Amazon 資源名稱 ( ARN ) 案例

在建置映像程序之外的執行個體中測試元件文件。若要取得需求清單，請參閱《Image Builder 使用指南》中的〈[元件管理員](#)〉。

成功驗證後，將組件添加到構建映像配置

確認自訂元件正在運作之後，請將其新增至 [Build 映像組態檔案](#)。

## 使用 `pcluster` 指令監視 Image Builder 程序，以協助偵錯

### [describe-image](#)

使用此命令可監視組建映像狀態。

### [list-image-log-streams](#)

使用此命令可取得可用來擷取記錄事件的記錄資料流 ID [get-image-log-events](#)。

### [get-image-log-events](#)

使用此命令可取得建置映像處理程序事件的記錄資料流。

例如，您可以使用以下命令來結尾構建圖像事件。

```
$ watch -n 1 'pcluster get-image-log-events -i <image-id> \
  --log-stream-name/1 <pcluster-version> \
  --query "events[*].message" | tail -n 50'
```

### [get-image-stack-events](#)

使用此命令可擷取映像產生器所建立之堆疊的影像堆疊事件。

## [export-image-logs](#)

使用此指令儲存影像記錄。

如需有關 AWS ParallelCluster 日誌和 Amazon 的詳細資訊 CloudWatch，請參閱[亞馬遜CloudWatch日誌構建映像日誌](#)和[Amazon CloudWatch 儀表](#)。

## 其他考量

### 新 AWS ParallelCluster 版本和自訂 AMI

如果您建置並使用自訂 AMI，則必須在每個新 AWS ParallelCluster 版本中重複建立自訂 AMI 時所用的步驟。

### 自定義引導操作

檢閱此[自訂引導動作](#)區段，以確定您要進行的修改是否可以編寫指令碼，並在 future 的 AWS ParallelCluster 版本中提供支援。

### 使用自訂 AMI

您可以在/[CustomAmi](#)和 [Scheduling/-Image/NameImage/CustomAmi](#)區段中的叢集配置中指定自訂 AMI。[SlurmQueues](#)

若要疑難排解自訂 AMI 驗證警告，請參閱[排解自訂 AMI 問題](#)。

## 使用隨需容量保留 (ODCR) 啟動執行個體

使用[隨需容量保留 \(ODCR\)](#)，您可以為特定可用區域中的叢集 Amazon EC2 執行個體預留容量。如此一來，您就可以獨立於 [Savings Plans](#) 或[地區預留執行個體](#)提供的帳單帳戶建立和管理容量保留。

您可以配置 open 或 targeted ODCR。「開啟 ODCR」會涵蓋符合 ODCR 屬性的所有實例。這些屬性包括執行個體類型、平台和可用區域。您必須在叢集配置中明確定義目標 ODCR。若要判斷是 open 否為 ODCR targeted，請執行 AWS CLI Amazon EC2 [describe-capacity-reservation](#) 命令。

您也可以稱呼為叢集置放群組 [隨選容量保留 \(CPG ODCR\)](#) 的叢集置放群組中建立 ODCR。

可以在資源群組中分組多個 ODC。這可以在群集配置文件中定義。如需有關資源群組的詳細資訊，請參閱[什麼是資源群組？](#)在《Resource Groups 與標籤使用指南》中。

## 使用 OCR 搭配使用 AWS ParallelCluster

AWS ParallelCluster 支持開放的 ODCR。使用開啟的 ODCR 時，您不需要在中指定任何內 AWS ParallelCluster 容。系統會自動為叢集選取執行個體。您可以指定現有的放置群組，也可以為您 AWS ParallelCluster 建立新的放置群組。

### 叢集配置中的 ODCR

從 AWS ParallelCluster 版本 3.3.0 開始，您可以在叢集組態檔中定義 ODC，而不需要指定 Amazon EC2 執行個體覆寫。

首先，您可以使用連結文件集中描述的方法來建立[容量保留](#)和[資源群組](#)。您必須使用這 AWS CLI 些方法來建立容量保留群組。如果使用 AWS Management Console，則只能建立以標籤為基礎或堆疊式資源群組。AWS ParallelCluster 或在啟動具有容量保留的執行個體 AWS CLI 時，不支援以標記為基礎和堆疊型資源群組。

建立容量保留區和資源群組之後，請在 [SlurmQueues/CapacityReservationTarget](#) 或 [SlurmQueuesComputeResources/CapacityReservationTarget](#) 中指定它們，如以下範例叢集配置所示。以有效#取代紅色亮度高的值。

```
Image:
  Os: os
HeadNode:
  InstanceType: head_node_instance
  Networking:
    SubnetId: public_subnet_id
  Ssh:
    KeyName: key_name
Scheduling:
  Scheduler: scheduler
SlurmQueues:
  - Name: queue1
    Networking:
      SubnetIds:
        - private_subnet_id
    ComputeResources:
      - Name: cr1
        Instances:
          - InstanceType: instance
        MaxCount: max_queue_size
        MinCount: max_queue_size
        Efa:
```



```
Enabled: true
CapacityReservationTarget:
  CapacityReservationResourceGroupArn: capacity_reservation_arn
```

過時/不建議-使用 Amazon EC2 執行個體覆寫的目標 ODCR

#### Warning

- 從 AWS ParallelCluster 版本 3.3.0 開始，我們不建議使用此方法。本節仍作為使用先前版本之實作的參考。
- 此方法與 Slurm 的多個實例類型分配不兼容。

在 3.1.1 中 AWS ParallelCluster 新增了對 targeted ODC 的 Support。在此版本中，引入了一種機制，該機制會覆寫 EC2 RunInstances 參數，並傳遞保留相關資訊，以用於中的每個已設定運算資源 AWS ParallelCluster。這種機制與 targeted ODCR 兼容。但是，當您使用 targeted ODCR 時，您必須指定 run-instances 覆寫組態。目標 ODCR 必須在 AWS CLI Amazon EC2 [run-instances](#) 命令中明確定義。判斷是否有 ODCR open 或 targeted 執行 AWS CLI Amazon EC2 命令 [describe-capacity-reservation](#)。

可以在資源群組中分組多個 ODC。這可用於執行執行個體覆寫，以同時鎖定多個 ODC。

如果您使用的是 targeted ODCR，您可以指定放置群組。但是，您也需要指定 run-instances 覆寫組態。

假設為您 AWS 建立了 targeted ODCR，或者您擁有一組特定的預留執行個體。然後，您就無法指定刊登位置群組。由所配置的規則 AWS 可能會與置放群組設定衝突。因此，如果您的申請需要安置組，請使用 [CPG ODCR](#)。在任何一種情況下，您也必須指定 run-instances 覆寫組態。

如果您使用的是 CPG ODCR，則必須指定 run-instances 覆寫配置，並且必須在叢集配置中指定相同的置放群組。

## 使用預留執行個體 AWS ParallelCluster

預留執行個體與容量保留 (ODCR) 不同。預留執行個體有 [2 種類型](#)。地區預留執行個體不保留容量。區域預留執行個體會在指定的可用區域中保留容量。

如果您有區域預留執行個體，則沒有容量保留，而且可能會出現容量不足錯誤。如果您有區域預留執行個體，則您有容量保留，但沒有可用來指定它們的 run-instances API 參數。

任何 AWS ParallelCluster 版本都支援預留執行個體。您不必在中指定任何內容，AWS ParallelCluster 而且會自動選取實體。

使用區域預留執行個體時，您可以省略叢集配置中的放置群組規格，以避免潛在的容量不足錯誤。

過時/不建議-針對**targeted**隨需容量保留 (ODCR) 使用 AWS ParallelCluster 3 中的 **RunInstances** 自訂

### Warning

- 從 AWS ParallelCluster 版本 3.3.0 開始，我們不建議使用此方法。本節仍作為使用先前版本之實作的參考。
- 此方法與 Slurm 的多個實例類型分配不兼容。

您可以為叢集佇列中設定的每個運算資源覆寫 Amazon EC2 RunInstances 參數。若要這麼做，請使用下列程式碼片段內容，在叢集的 head 節點上建立 `/opt/slurm/etc/pcluster/run_instances_overrides.json` 檔案：

- `${queue_name}` 是您要套用覆寫項目的佇列名稱。
- `${compute_resource_name}` 是您要套用覆寫的計算資源。
- `${overrides}` 是任意 JSON 物件，其中包含要用於佇列和執行個體類型之特定組合的 RunInstances 覆寫清單。覆寫語法需要遵循與 [run\\_instance](#) boto3 呼叫中所記錄的相同規格。

```
{
  "${queue_name}": {
    "${compute_resource_name}": {
      ${overrides}
    },
    ...
  },
  ...
}
```

例如，下列 JSON 會設定 `group_arn` 要用於和中設定的 `p4d.24xlarge` 執行個體的 ODCR 群組。 `my-queue my-compute-resource`

```
{
  "my-queue": {
```

```

    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationResourceGroupArn": "group_arn"
        }
      }
    }
  }
}

```

產生此 JSON 檔案之後，負責叢集調整規模的 AWS ParallelCluster 精靈會自動使用執行個體啟動時的覆寫設定。若要確認指定的參數正在用於執行處理啟動設定，請查看下列記錄檔：

- /var/log/parallelcluster/clustermgtd(靜態容量)
- /var/log/parallelcluster/slurm\_resume.log(適用於動態容量)

如果參數正確，您會發現包含下列項目的記錄項目：

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

過時/不建議-建立具有**targeted**隨選容量保留 (ODCR) 的叢集

#### Warning

- 從 AWS ParallelCluster 版本 3.3.0 開始，我們不建議使用此方法。本節仍作為使用先前版本之實作的參考。
- 此方法與[使用 Slurm 進行多個實例類型分配](#)。

1. 建立資源群組，以群組產能。

```
$ aws resource-groups create-group --name EC2CRGroup \
  --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
  '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

#### Note

資源群組不支援由其他帳號共用的資源。

如果目標 ODCR 由另一個帳戶共用，則不需要建立資源群組。在步驟 3 中使用 `CapacityReservationId` 而非資源群組。

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

# Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
  "my-queue": {
    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationId": "cr-abcdef01234567890"
        }
      }
    }
  }
}
EOF
```

將產能保留新增至資源群組。每次建立新的 ODCR 時，請將其新增至群組保留區。以您 `ACCOUNT_ID` 的帳戶 ID、您 `PLACEHOLDER_CAPACITY_RESERVATION` 的容量保留識別碼以及您 `REGION_ID` 的 AWS 區域 識別碼 (例如 us-east-1) 取代。

```
$ aws resource-groups group-resources --region REGION_ID --group EC2CRGroup \
  --resource-arns arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-
  reservation/PLACEHOLDER_CAPACITY_RESERVATION
```

在您的本機電腦上建立原則文件。 `ACCOUNT_ID` 用您的帳戶 ID 和您 `REGION_ID` 的 AWS 區域 ID 替換 (例如 us-east-1)。

```
cat > policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RunInstancesInCapacityReservation",
      "Effect": "Allow",
```

```

        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-reservation/*",
            "arn:aws:resource-groups:REGION_ID:ACCOUNT_ID:group/*"
        ]
    }
]
}
EOF

```

2. AWS 帳戶 使用您建立的 json 檔案建立 IAM 政策。

```

$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-
document file://policy.json

```

3. 在執行個體本機上建立下列安裝後指令碼，並將其命名 **postinstall.sh**。

**ACCOUNT\_ID**用您的 AWS 帳戶 身份證和您**REGION\_ID**的 AWS 區域 身份證替換 ( 例如 us-east-1 )。

```

#!/bin/bash
set -e

# Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
    "my-queue": {
        "my-compute-resource": {
            "CapacityReservationSpecification": {
                "CapacityReservationTarget": {
                    "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:REGION_ID:ACCOUNT_ID:group/EC2CRGroup"
                }
            }
        }
    }
}
EOF

```

將檔案上傳到 Amazon S3 儲存貯體。將 **S3\_NAME ##### S3 #####**名稱。

```

$ aws s3 mb s3://S3_NAME_BUCKET

```

```
aws s3 cp postinstall.sh s3://S3_NAME_BUCKET/postinstall.sh
```

#### 4. 建立本機叢集配置，並以您自己的值取代預留位置。

```
Region: REGION_ID
Image:
  Os: alinux2
HeadNode:
  InstanceType: c5.2xlarge
  Ssh:
    KeyName: YOUR_SSH_KEY
  Iam:
    S3Access:
      - BucketName: S3_NAME_BUCKET
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::ACCOUNT_ID:policy/RunInstancesCapacityReservation
## This post-install script is executed after the node is configured.
## It is used to install scripts at boot time and specific configurations
## In the script below we are overriding the calls to RunInstance to force
## the provisioning of our my-queue partition to go through
## the On-Demand Capacity Reservation
CustomActions:
  OnNodeConfigured:
    Script: s3://S3_NAME_BUCKET/postinstall.sh
Networking:
  SubnetId: YOUR_PUBLIC_SUBNET_IN_TARGET_AZ

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: my-queue
      ComputeResources:
        - MinCount: 0
          MaxCount: 100
          InstanceType: p4d.24xlarge
          Name: my-compute-resource
          Efa:
            Enabled: true
      Networking:
        ## PlacementGroup:
        ##   Enabled: true ## Keep PG disabled if using targeted ODCR
        SubnetIds:
          - YOUR_PRIVATE_SUBNET_IN_TARGET_AZ
```

## 5. 建立叢集。

使用下列指令建立叢集。*cluster-config.yaml*用您的配置文件名稱，*cluster-dl*用您的集群名稱替換為 *REGION\_ID* 與您的區域 ID ( 例如 , us-east-1 ) 。

```
$ pcluster create-cluster --cluster-configuration cluster-config.yaml --cluster-name cluster-dl --region REGION_ID
```

建立叢集之後，安裝後指令碼會在 head 節點中執行。指令碼會建立 *run\_instances\_overrides.json* 檔案並覆寫呼叫，RunInstances 以強制佈建磁碟分割透過隨選容量保留。

負責叢集擴展的 AWS ParallelCluster 精靈會針對已啟動的新執行個體自動使用此設定。若要確認指定的參數是用來佈建執行個體，您可以查看下列記錄檔：

- /var/log/parallelcluster/clustermgtd(靜態容量-[MinCount](#) > 0)
- /var/log/parallelcluster/slurm\_resume.log(適用於動態容量)

如果參數正確，您會發現包含以下內容的日誌項目。

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

### 更新RunInstances覆寫

您可以隨時更新產生的 JSON 組態，而不必停止運算叢集。套用變更之後，所有新執行個體都會以更新的組態啟動。如果您需要將更新的設定套用至執行中的節點，請強制執行個體終止來回收節點，然後等待 AWS ParallelCluster 取代這些節點。您可以透過從 Amazon EC2 主控台終止執行個體 AWS CLI，或者在 DOWN 或 DRAIN 狀態中設定 Slurm 節點來執行此操作。

使用以下命令將 Slurm 節點設置為 DOWN 或 DRAIN。

```
$ scontrol update nodename=my-queue-dy-my-compute-resource-1 state=down  
reason=your_reason  
scontrol update nodename=my-queue-dy-my-compute-resource-1 state=drain  
reason=your_reason
```

## 使用容量區塊 (CB) 啟動執行個體

AWS ParallelCluster 支援 [Machine Learning 的隨需容量保留 \(ODCR\) 和容量區塊 \(CB\)](#)。與 ODCR 不同，CB 可以擁有 future 的開始時間，並且具有時間限制。如需有關使用 ODCR 啟動的詳細資訊，請參閱[使用隨需容量保留 \(ODCR\) 啟動執行個體](#)。

### 使用 CB 搭配使用 AWS ParallelCluster

若要設定新叢集或現有叢集以使用 CB，您必須先在 AWS 帳戶中擁有有效的 CB。您可以按照官方文件使用 AWS Management Console AWS Command Line Interface、或 SDK 來尋找和購買可用的 CB。擁有有效的 CB 後，您可以在 AWS ParallelCluster 組態檔案中設定 CB Amazon 資源名稱 (ARN) 和相關參數。如需詳細資訊，請參閱[尋找和購買容量區塊 \(CB\)](#)

#### 叢集配置中的 CB

若要將 CB 用於特定佇列，您需要使用CapacityReservationId參數。將其設定為現有的 CB ID。您可以從您用來建立 CB 的 AWS Management Console AWS CLI、或 SDK 取得 CB ARN。

您必須CapacityType = CAPACITY\_BLOCK為要使用 CB 的佇列進行設定。將其設定為運算資源InstanceType的 (與 CB 相同的 Amazon 彈性運算雲端執行個體類型)。

如果CapacityReservationId在計算資源層級指定，則InstanceType為選用性，因為系統會自動從保留區擷取該資源。

使用時CapacityType = CAPACITY\_BLOCK，MaxCount必須等於或大於 MinCount 0，因為屬於 CB 保留項目一部分的所有執行個體都會作為靜態節點進行管理。

在叢集建立時，頭節點會等待所有靜態節點準備就緒，然後才能成功建立叢集。但是，使用時CapacityType = CAPACITY\_BLOCK，屬於與之計算資源一部分的節點不會被考慮用於此檢查。即使並非所有配置都處於活動狀態，也將創建集群。

下列組態檔案程式碼片段顯示要在 AWS ParallelCluster 組態檔案中啟用的必要參數。

```
SlurmQueues:
- Name: string
  CapacityType: CAPACITY_BLOCK
  InstanceType: String (EC2 Instance type of the CB)
  MinCount: integer (<= total capacity of the CB)
  MaxCount: integer (equal to MinCount)
  ComputeResources:
  - Name: string
```



```
CapacityReservationTarget:  
  CapacityReservationId: String (CB id)
```

## 如何 AWS ParallelCluster 使用容量區塊 (CB)

AWS ParallelCluster 以特殊的方式管理與相關的靜態節點。AWS ParallelCluster 即使 CB 尚未處於作用中狀態，仍會建立叢集，而當 CB 處於作用中狀態時，執行個體也會自動啟動。

與計算資源相關聯且尚未啟動的 Slurm 節點對應的節點會保持維護狀態，直到它們到達 CB 開始時間為止。Slurm 節點保持在保留/維護狀態，並與 slurm 管理員使用者相關聯。這表示他們可以接受工作，但工作會一直保留在中，pending 直到移除保留項目為止。

AWS ParallelCluster 會自動更新保留 Slurm 留區，並將相關 CB 節點置於維護中 (對應 CB 狀態)。CB 處於作用中狀態時，會移除 Slurm 保留區、節點啟動，並可供擱置的工作或新工作提交使用。

到達 CB 結束時間時，節點會移回保留/維護狀態。當 CB 不再處於作用中狀態且執行個體終止時，使用者可以將作業重新提交/重新報告至新佇列/運算資源。

## AMI 修補和 Amazon EC2 實例替換

若要確保所有動態啟動的叢集運算節點都以一致的方式運作，請 AWS ParallelCluster 停用叢集執行個體自動作業系統更新。此外，還會針對每個版本的 AWS ParallelCluster 及其關聯的 CLI 建置一組特定的 AWS ParallelCluster AMI。這組特定的 AMI 保持不變，只有它們建立的 AWS ParallelCluster 版本才支援它們。AWS ParallelCluster 已發行版本的 AMI 不會更新。

不過，由於緊急的安全性問題，客戶可能想要新增修補程式至這些 AMI，然後使用修補的 AMI 更新其叢集。這與「[AWS ParallelCluster 共同責任模型](#)」一致。

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

```
$ pcluster version  
$ pcluster list-official-images
```

AWS ParallelCluster 頭節點是靜態實例，您可以手動更新它。從 3.0.0 AWS ParallelCluster 版開始，完全支援頭節點的重新啟動和重新啟動。

如果您的執行個體有暫時性的執行個體存放區，您必須記得在手動更新之前儲存執行個體存放區資料。[如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的 `HeadNodeLocalStorage/EphemeralVolume` 叢集組態和執行個體存放區磁碟區的執行個體類型。](#)

運算節點是暫時執行個體。默認情況下，您只能從頭節點訪問它們。從 3.0.0 AWS ParallelCluster 版開始，您可以透過修改 [Scheduling//SlurmQueuesImage/CustomAmi](#) 參數並執行 `pcluster update-cluster` 命令來更新與計算執行個體相關聯的 AMI，在停止運算叢集之後：[pcluster update-compute-fleet](#)

```
$ pcluster update-compute-fleet-status --status STOP_REQUESTED
```

您可以使用下列其中一種方法，為運算節點自動建立更新的自訂 AMI：

- 使用命 `pcluster build-image` 令搭配更新的 [Build/ParentImage](#)。
- 使用 [Build/UpdateOsPackages/Enabled](#)：運行構建 `true`。

## 頭節點實例更新或更換

在某些情況下，您可能需要重新啟動或重新啟動頭節點。例如，當您手動更新作業系統，或者有 [AWS 執行個體排定淘汰而強制執行頭節點執行個體](#) 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，您可以隨時停止並重新啟動它。在排程淘汰的情況下，啟動停止的執行個體會將其移轉為使用新硬體。

同樣地，您也可以手動停止和啟動沒有執行個體存放區的執行個體。對於這種情況以及沒有臨時磁碟區的執行個體的其他情況，請繼續執行。[停止並啟動叢集的頭節點](#)

如果您的執行個體具有暫時性磁碟機且已停止，則執行個體存放區中的資料將會遺失。您可以從執行個體儲存磁碟區中的表格中，判斷用於標頭節點的執行個體類型是否具有 [執行個體存放區](#)。

## 從暫時驅動器保存數據

從 3.0.0 AWS ParallelCluster 版開始，每個執行個體類型都完全支援主節點重新啟動和重新啟動。但是，如果執行個體具有暫時磁碟機，則其資料將會遺失。請遵循以下步驟，在頭節點重新啟動或重新啟動之前保留資料。

要檢查您是否有需要保留的數據，請查看 [EphemeralVolume/MountDir](#) 文件夾中的內容（ /scratch 默認情況下）。

您可以將資料傳輸到根磁碟區或連接到叢集的共用儲存系統，例如 Amazon FSx、Amazon EFS 或 Amazon EBS。請注意，資料傳輸到遠端儲存可能會產生額外費用。

儲存資料後，請繼續執行 [停止並啟動叢集的頭節點](#)。

## 停止並啟動叢集的頭節點

1. 確認叢集中沒有任何執行中的工作。

使用Slurm排程器時：

- 如果未指定`sbatch--no-requeue`選項，則會重新計算執行中的工作。
- 如果指定了該`--no-requeue`選項，則執行作業會失敗。

2. 要求叢集運算叢集停止：

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOP_REQUESTED",
  ...
}
```

3. 等到運算叢集狀態為STOPPED：

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOPPED",
  ...
}
```

4. 若要透過作業系統重新啟動或執行個體重新啟動進行手動更新，您可以使用 AWS Management Console 或 AWS CLI。以下是使用的範例 AWS CLI。

```
# Retrieve head node instance id
$ pcluster describe-cluster --cluster-name cluster-name --status STOP_REQUESTED
{
  "headNode": {
    "instanceId": "i-1234567890abcdef0",
    ...
  },
  ...
}
# stop and start the instance
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
  "StoppingInstances": [
    {
      "CurrentState": {
```

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

## 5. 啟動叢集運算叢集：

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
  "status": "START_REQUESTED",
  ...
}
```

## 作業系統

AWS ParallelCluster 支持 Amazon Linux 2, Amazon Linux 2023, CentOS 7, Ubuntu 22.04, 紅帽企業 Linux 8 (RHEL8), 紅帽企業 Linux 8 (), 紅帽企業 Linux 9 (RHEL9), 和洛基 9. AWS ParallelCluster 為選

定的操作系統提供預先構建的 AMI，有關 AMI 的更多詳細信息，AWS ParallelCluster 請參閱。[Image 區段](#)

## 作業系統考量

### Ubuntu

Ubuntu 2204 需要更安全的 SSH 密鑰，默認情況下不支持 RSA 密鑰。請產生一個 ed25519 金鑰，並將其用於叢集建立。

Ubuntu 2204 無法更新為最新的核心，因為該核心沒有 Fx 用戶端。

### RHEL 8

RedHat 從 3.6.0 版本開始增加了企業 AWS ParallelCluster 版本 8.7 ( 如果您將叢集設定為使用 rhel8，則任何執行個體類型的隨需費用都會高於您將叢集設定為使用其他支援的作業系統時的隨需費用。

[如需有關定價的詳細資訊，請參閱隨需定價以及 Amazon 彈性運算雲端上的 RHEL 如何提供和定價？](#)

。

### 洛奇 8 號

AWS ParallelCluster 3.8.0 支持洛基 Linux 8，但預建的落基 Linux 8 AMI ( 對於 x86 和 ARM 架構 ) 是不可用的。AWS ParallelCluster 3.8.0 支援使用該[CustomAmi](#)屬性使用自訂 AMI 建立叢集，使用岩石 Linux 8。如需有關建置自訂 AMI 的詳細資訊，請參閱[AWS ParallelCluster AMI 定制](#)。

[要從基礎岩 Linux 8 AMI 構建自定義 AMI，您可以考慮訂閱岩基 Linux 8 AMI 可在 Marketplace 上 AWS。](#)請務必檢閱 AWS Marketplace 上的落基 Linux 8 AMI 的定價和訂閱費用。或者，您也可以使用[官方的洛基 Linux 8 AMI 作為您的基礎 AMI](#)。

### 人口普查 7

[Gdrcopy](#) 已從其作業系統支援矩陣中移除了 Centos7。這意味著 gdrcopy 2.3.1 是支持此操作系統的最新版本。你必須固定英偉達和博德鏡版本的 Centos7，因為最新的 NVIDIA 開源驅動程序版本 ( OpenRM 即 535.129.03 + ) 與此版本的格式複製不兼容。從 ParallelCluster 3.8.0 開始，我們的官方普托斯 7 AMI 將與光電複製版 2.3.1 和 NVIDIA 驅動程序 535.129.03 發布。

### 搖滾樂 9

AWS ParallelCluster 3.9.0 支持洛基 Linux 9，但預建的落基 Linux 9 AMI ( 對於 x86 和 ARM 架構 ) 是不可用的。AWS ParallelCluster 3.9.0 支援使用該[CustomAmi](#)屬性使用自訂 AMI 建立叢集使用岩石

Linux 9。如需有關建置自訂 AMI 的詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。要從基礎洛基 Linux 9 AMI 構建自定義 AMI，您還可以使用 [官方落基 Linux 9 AMI 作為您的基礎 AMI](#)。如果基礎 AMI 沒有最新的內核，則自定義洛基 Linux 9 AMI 構建可能會失敗。要在構建 AMI 之前升級內核：

- [從這裡使用一個岩石 9 AMI ID 啟動一個實例](https://rockylinux.org/cloud-images/)：https://rockylinux.org/cloud-images/
- ssh 進入實例並運行以下命令：sudo yum -y update
- 從執行個體建立映像檔以做為 ParentImage

# 參考資料 AWS ParallelCluster

## 主題

- [AWS ParallelCluster CLI 命令](#)
- [組態檔案](#)
- [AWS ParallelCluster API 參考資料](#)
- [AWS ParallelClusterPython 函式庫](#)

## AWS ParallelCluster CLI 命令

`pcluster` 是主要的 AWS ParallelCluster CLI 指令。您可 `pcluster` 以用來啟動和管理中的 HPC 叢集，以 AWS 雲端及建立和管理自訂 AMI 映像。

`pcluster3-config-converter` 用於將 AWS ParallelCluster 版本 2 格式的叢集配置轉換為 AWS ParallelCluster 版本 3 格式。

```
pcluster [-h] ( build-image | configure |
    create-cluster | dcv-connect |
    delete-cluster | delete-cluster-instances | delete-image |
    describe-cluster | describe-cluster-instances |
    describe-compute-fleet | describe-image |
    export-cluster-logs | export-image-logs |
    get-cluster-log-events | get-cluster-stack-events |
    get-image-log-events | get-image-stack-events |
    list-cluster-log-streams | list-clusters |
    list-images | list-image-log-streams | list-official-images |
    ssh | update-cluster |
    update-compute-fleet | version ) ...
pcluster3-config-converter [-h] [-t CLUSTER_TEMPLATE]
    [-c CONFIG_FILE]
    [--force-convert]
    [-o OUTPUT_FILE]
```

## 主題

- [pcluster](#)
- [pcluster3-config-converter](#)

# pcluster

pcluster 是主要的 AWS ParallelCluster CLI 指令。您可 pcluster 以用來啟動和管理中的 AWS 雲端 HPC 叢集。

pcluster 將指令的記錄寫入中的 `pcluster.log.#檔案/home/user/.parallelcluster/`。如需詳細資訊，請參閱 [pclusterCLI 記錄檔](#)。

若要使用 pcluster，您必須擁有具有執行該角色所需 [許可](#) 的 IAM 角色。

```
pcluster [-h]
```

## 引數

### pcluster *command*

可能的選擇：[build-image](#)[configure](#)[create-cluster](#)[dcv-connect](#)[delete-cluster](#)[delete-cluster-instances](#)[delete-image](#)[describe-cluster](#)[describe-cluster-instances](#)[describe-compute-fleet](#)[describe-image](#)[export-cluster-log](#)[export-image-log](#)[get-cluster-log](#)[get-cluster-stack-events](#)[get-image-log](#)[get-image-stack-events](#)[list-clusters](#)[list-cluster-log-streams](#)[list-images](#)[list-image-log-streams](#)[list-official-images](#)[ssh](#)[update-cluster](#)[update-compute-fleet](#)[version](#)

子命令：

## 主題

- [pcluster build-image](#)
- [pcluster configure](#)
- [pcluster create-cluster](#)
- [pcluster dcv-connect](#)
- [pcluster delete-cluster](#)
- [pcluster delete-cluster-instances](#)
- [pcluster delete-image](#)
- [pcluster describe-cluster](#)



- [pcluster describe-cluster-instances](#)
- [pcluster describe-compute-fleet](#)
- [pcluster describe-image](#)
- [pcluster export-cluster-logs](#)
- [pcluster export-image-logs](#)
- [pcluster get-cluster-log-events](#)
- [pcluster get-cluster-stack-events](#)
- [pcluster get-image-log-events](#)
- [pcluster get-image-stack-events](#)
- [pcluster list-clusters](#)
- [pcluster list-cluster-log-streams](#)
- [pcluster list-images](#)
- [pcluster list-image-log-streams](#)
- [pcluster list-official-images](#)
- [pcluster ssh](#)
- [pcluster update-cluster](#)
- [pcluster update-compute-fleet](#)
- [pcluster version](#)

## pcluster build-image

在指定的區域中建立自訂 AWS ParallelCluster 映像檔。

```
pcluster build-image [-h]
                    --image-configuration IMAGE_CONFIGURATION
                    --image-id IMAGE_ID
                    [--debug]
                    [--dryrun DRYRUN]
                    [--query QUERY]
                    [--region REGION]
                    [--rollback-on-failure ROLLBACK_ON_FAILURE]
                    [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
                    [--validation-failure-level {INFO,WARNING,ERROR}]
```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster build-image`。

### **--image-configuration, -c *IMAGE\_CONFIGURATION***

將映像組態檔指定為 YAML 文件。

### **--image-id, -i *IMAGE\_ID***

指定將要建置之影像的 ID。

### **--debug**

開啟偵錯記錄

### **--dryrun *DRYRUN***

何時 `true`，命令會執行驗證，而不建立任何資源。您可以使用它來驗證映像配置。(預設為 `false`。)

### **--query *QUERY***

JMESPath 查詢在輸出上執行。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用影像組態檔案中的「區域」([Region](#)) 設定、`AWS_DEFAULT_REGION`環境變數、`~/.aws/config`檔案區[`default`]段中的 `region` 設定或 `--region` 參數來指定。

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

何時 `true`，會在失敗時自動啟動映像堆疊復原。(預設為 `false`。)

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

識別要抑制的一或多個組態驗證程式。

格式: `(ALL|type:[A-Za-z0-9]+)`

### **--validation-failure-level {INFO,WARNING,ERROR}**

指定造成建立失敗的最低驗證層級。(預設為 `ERROR`。)

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster build-image --image-configuration image-config.yaml --image-id custom-  
alinux2-image  
{  
  "image": {  
    "imageId": "custom-alinux2-image",  
    "imageBuildStatus": "BUILD_IN_PROGRESS",  
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",  
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
custom-alinux2-image/1234abcd-56ef-78gh-90ij-abcd1234efgh",  
    "region": "us-east-1",  
    "version": "3.1.2"  
  }  
}
```

#### Warning

`pcluster build-image` 使用預設的 VPC。如果已刪除預設 VPC (可能是使用 AWS Control Tower 或 AWS 登陸區域)，則必須在映像設定檔中指定子網路 ID。如需詳細資訊，請參閱 [SubnetId](#)。

## pcluster configure

啟動第 3 AWS ParallelCluster 版的互動式設定精靈。如需詳細資訊，請參閱 [使用 AWS ParallelCluster 命令行界面配置和創建集群](#)。

```
pcluster configure [-h]  
    --config CONFIG  
    [--debug]  
    [--region REGION]
```

### 命名參數

#### **-h, --help**

顯示的說明文字 `pcluster configure`。

#### **--config *CONFIG***

輸出生成的配置文件的路徑。

**--debug**

開啟偵錯記錄

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

指定 AWS 區域 要使用的。必須使用影像組態檔案中的「區域」([Region](#)) 設定、AWS\_DEFAULT\_REGION環境變數、檔~/.aws/config案區[default]段中的region設定或--region參數來指定 Region。

**pcluster create-cluster**

建立 AWS ParallelCluster 叢集。

```
pcluster create-cluster [-h]
                        --cluster-configuration CLUSTER_CONFIGURATION
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--dryrun DRYRUN]
                        [--query QUERY]
                        [--region REGION]
                        [--rollback-on-failure ROLLBACK_ON_FAILURE]
                        [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]

                        [--validation-failure-level {INFO,WARNING,ERROR}]
```

**命名參數****-h, --help**

顯示的說明文字pcluster create-cluster。

**--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

指定 YAML 叢集配置檔案。

**--cluster-name, -n *CLUSTER\_NAME***

指定要建立的叢集名稱。

名稱必須以字母字元開頭。名稱最多可包含 60 個字元。如果啟用Slurm會計，名稱最多可包含 40 個字元。

有效字元：a-z、A-Z、0-9 和-(連字號)。

**--debug**

啟用偵錯記錄。

**--dryrun *DRYRUN***

何時true，命令會執行驗證，而不建立任何資源。您可以使用它來驗證叢集配置。(預設為false。)

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用叢集配置檔案中的[Region](#)設定、AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

**--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

何時true，會在失敗時自動啟動叢集堆疊復原。(預設為true。)

**--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

識別要抑制的一或多個組態驗證程式。

格式:(ALL| 類型:[A-Za-z0-9]+)

**--validation-failure-level {INFO,WARNING,ERROR}**

指定造成建立失敗的最低驗證層級。(預設為ERROR。)

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster create-cluster -c cluster-config.yaml -n cluster-v3
{
  "cluster": {
    "clusterName": "cluster-v3",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
    "region": "us-east-1",
    "version": "3.1.4",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

```
}
```

## pcluster dcv-connect

允許通過使用 NICE DCV 通過交互式會話連接到頭節點。

```
pcluster dcv-connect [-h]
                    --cluster-name CLUSTER_NAME
                    [--debug]
                    [--key-path KEY_PATH]
                    [--region REGION]
                    [--show-url]
```

### 命名參數

#### **-h, --help**

顯示的說明文字 pcluster dcv-connect。

#### **--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

#### **--debug**

啟用偵錯記錄。

#### **--key-path *KEY\_PATH***

指定用於連線的 SSH 金鑰路徑。

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

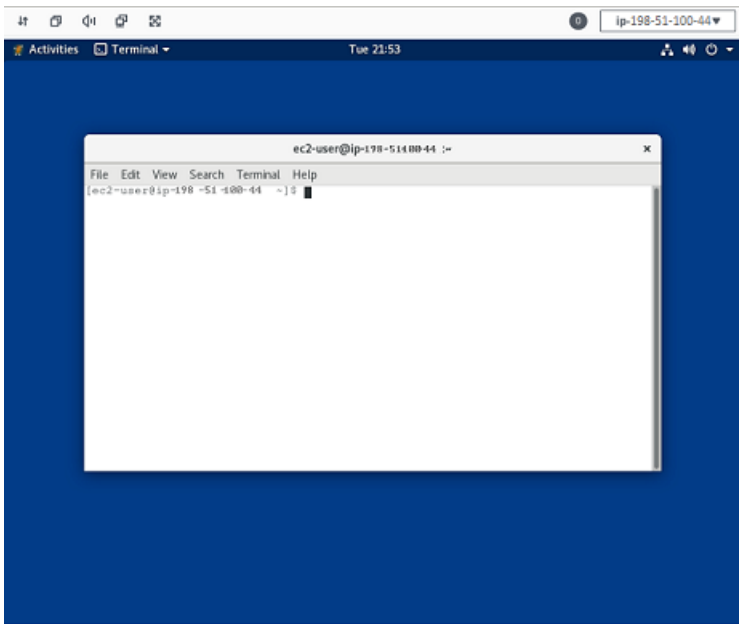
指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

#### **--show-url**

列印將用於 DCV 連線的 URL 並結束。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster dcv-connect -n cluster-3Dcv -r us-east-1 --key-path /home/user/.ssh/key.pem
```



## pcluster delete-cluster

開始刪除叢集。

```
pcluster delete-cluster [-h]
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

命名參數

**-h, --help**

顯示的說明文字 pcluster delete-cluster。

**--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

**--debug**

啟用偵錯記錄。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案區[default]段中的region設定或--region參數來指定 Region。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster delete-cluster -n cluster-v3
{
  "cluster": {
    "clusterName": "cluster-v3",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
    "region": "us-east-1",
    "version": "3.1.4",
    "clusterStatus": "DELETE_IN_PROGRESS"
  }
}
```

**pcluster delete-cluster-instances**

啟動所有叢集運算節點的強制終止。這不適用於 AWS Batch 叢集。

```
pcluster delete-cluster-instances [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--force FORCE]
    [--query QUERY]
    [--region REGION]
```

命名參數

**-h, --help**

顯示的說明文字pcluster delete-cluster-instances。

**--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。



**--debug**

啟用偵錯記錄。

**--force *FORCE***

何時true，會忽略驗證錯誤來強制刪除。(預設為false。)

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

```
$ pcluster delete-cluster-instances -n cluster-v3
```

## pcluster delete-image

開始刪除自訂 AWS ParallelCluster 影像。

```
pcluster delete-image [-h]
                    --image-id IMAGE_ID
                    [--debug]
                    [--force FORCE]
                    [--query QUERY]
                    [--region REGION]
```

### 命名參數

**-h, --help**

顯示的說明文字pcluster delete-image。

**--image-id, -i *IMAGE\_ID***

指定要刪除之影像的 ID。

**--debug**

啟用偵錯記錄。

**--force *FORCE***

何時true，如果有實例使用 AMI 或共享 AMI，則強制刪除。(預設為false。)

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster delete-image --image-id custom-alinux2-image
{
  "image": {
    "imageId": "custom-alinux2-image",
    "imageBuildStatus": "DELETE_IN_PROGRESS",
    "region": "us-east-1",
    "version": "3.1.4"
  }
}
```

## pcluster describe-cluster

取得叢集的詳細資訊。

```
pcluster describe-cluster [-h]
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

命名參數

**-h, --help**

顯示的說明文字pcluster describe-cluster。

**--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

**--debug**

啟用偵錯記錄。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

描述叢集詳細資料：

```
$ pcluster describe-cluster -n cluster-v3
{
  "creationTime": "2022-07-12T17:19:16.101Z",
  "headNode": {
    "launchTime": "2022-07-12T17:22:21.000Z",
    "instanceId": "i-1234567890abcdef0",
    "publicIpAddress": "198.51.100.44",
    "instanceType": "t2.micro",
    "state": "running",
    "privateIpAddress": "192.0.2.0.196"
  },
  "loginNodes": {
    "status": "active",
    "address": "8af2145440569xyz.us-east-1.amazonaws.com",
    "scheme": "internet-facing|internal",
    "healthyNodes": 3,
    "unhealthyNodes": 0
  },
  "version": "3.1.4",
  "clusterConfiguration": {
    "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
  },
  "tags": [
    {
      "value": "3.1.4",
      "key": "parallelcluster:version"
    }
  ]
}
```

```
],
"cloudFormationStackStatus": "CREATE_COMPLETE",
"clusterName": "cluster-v3",
"computeFleetStatus": "RUNNING",
"cloudFormationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
"lastUpdatedTime": "2022-07-12T17:19:16.101Z",
"region": "us-east-1",
"clusterStatus": "CREATE_COMPLETE"
}
```

用describe-cluster於擷取叢集配置：

```
$ curl -o - - $(pcluster describe-cluster -n cluster-v3 --query clusterConfiguration.url
| xargs echo)
Region: us-east-1
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: adpc
  Iam:
    S3Access:
      - BucketName: cluster-v3-bucket
        KeyName: logs
        EnableWriteAccess: true
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue1
        ComputeResources:
          - Name: t2micro
            InstanceType: t2.micro
            MinCount: 0
            MaxCount: 10
        Networking:
          SubnetIds:
            - subnet-021345abcdef6789
```

## pcluster describe-cluster-instances

說明叢集中的執行個體。

```
pcluster describe-cluster-instances [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--next-token NEXT_TOKEN]
    [--node-type {HeadNode,ComputeNode,LoginNode}]
    [--query QUERY]
    [--queue-name QUEUE_NAME]
    [--region REGION]
```

### 命名參數

#### **-h, --help**

顯示的說明文字pcluster describe-cluster-instances。

#### **--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

#### **--debug**

啟用偵錯記錄。

#### **--next-token *NEXT\_TOKEN***

指定要用於分頁請求的權杖。

#### **--node-type {HeadNode,ComputeNode,LoginNode}**

指定要列出的節點類型。支援的值為HeadNode、ComputeNode和LoginNode。如果未指定此參數HeadNode，則會描述ComputeNode和執行LoginNode個體。

#### **--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

#### **--queue-name *QUEUE\_NAME***

指定要列出的佇列名稱。如果未指定此參數，則會說明所有佇列中的執行個體。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster describe-cluster-instances -n cluster-v3
{
  "instances": [
    {
      "launchTime": "2022-07-12T17:22:21.000Z",
      "instanceId": "i-1234567890abcdef0",
      "publicIpAddress": "198.51.100.44",
      "instanceType": "t2.micro",
      "state": "running",
      "nodeType": "HeadNode",
      "privateIpAddress": "192.0.2.0.196"
    },
    {
      "launchTime": "2022-07-12T17:37:42.000Z",
      "instanceId": "i-021345abcdef6789",
      "queueName": "queue1",
      "publicIpAddress": "198.51.100.44",
      "instanceType": "t2.micro",
      "state": "pending",
      "nodeType": "ComputeNode",
      "privateIpAddress": "192.0.2.0.196"
    }
  ]
}
```

## pcluster describe-compute-fleet

描述運算叢集的狀態。

```
pcluster describe-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--query QUERY]
    [--region REGION]
```

命名參數

**-h, --help**

顯示的說明文字 `pcluster describe-compute-fleet`。

**--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

**--debug**

啟用偵錯記錄。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
  "status": "RUNNING",
  "lastStatusUpdateTime": "2022-07-12T17:24:26.000Z"
}
```

## pcluster describe-image

獲取有關圖像的詳細信息。

```
pcluster describe-image [-h]
                        --image-id IMAGE_ID
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

命名參數

**-h, --help**

顯示的說明文字pcluster describe-image。

**--image-id, -i *IMAGE\_ID***

指定影像的識別碼。

**--debug**

啟用偵錯記錄。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster describe-image --image-id custom-alinux2-image
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678-v1-do-not-delete.../configs/image-
config.yaml"
  },
  "imageId": "custom-alinux2-image",
  "creationTime": "2022-04-05T20:23:07.000Z"
  "imageBuildStatus": "BUILD_COMPLETE",
  "region": "us-east-1",
  "ec2AmiInfo": {
    "amiName": "custom-alinux2-image 2022-04-05T19-55-22.518Z",
    "amiId": "ami-1234abcd5678efgh",
    "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.268-205.500.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.14.2-1.amzn2.x86_64, dcv-2021.3.11591-1.el7.x86_64, slurm-21-08-6-1",
    "state": "AVAILABLE",
    "tags": [
      {
        "value": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
        "key": "Ec2ImageBuilderArn"
      },
      {
        "value": "parallelcluster-1234abcd5678efgh-v1-do-not-delete",
        "key": "parallelcluster:s3_bucket"
      },
      {
        "value": "custom-alinux2-image",
```



```
    "key": "parallelcluster:image_name"
  },
  {
    "value": "available",
    "key": "parallelcluster:build_status"
  },
  {
    "value": "s3://parallelcluster-1234abcd5678efgh-v1-do-not-delete/
parallelcluster/3.1.2/images/custom-alinux2-image-1234abcd5678efgh/configs/image-
config.yaml",
    "key": "parallelcluster:build_config"
  },
  {
    "value": "Amazon EC2 Image Builder",
    "key": "CreatedBy"
  },
  {
    "value": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-alinux2-image",
    "key": "parallelcluster:build_log"
  },
  {
    "value": "4.14.268-205.500.amzn2.x86_64",
    "key": "parallelcluster:kernel_version"
  },
  {
    "value": "arn:aws:imagebuilder:us-east-1:444455556666:image/amazon-linux-2-
x86/2022.3.16/1",
    "key": "parallelcluster:parent_image"
  },
  {
    "value": "3.1.2",
    "key": "parallelcluster:version"
  },
  {
    "value": "0.5.14",
    "key": "parallelcluster:munge_version"
  },
  {
    "value": "21-08-6-1",
    "key": "parallelcluster:slurm_version"
  },
  {
    "value": "2021.3.11591-1.el7.x86_64",
```

```
    "key": "parallelcluster:dcv_version"
  },
  {
    "value": "alinux2-image",
    "key": "parallelcluster:image_id"
  },
  {
    "value": "3.2.3",
    "key": "parallelcluster:pmix_version"
  },
  {
    "value": "parallelcluster/3.7.0/images/alinux2-image-abcd1234efgh56781234",
    "key": "parallelcluster:s3_image_dir"
  },
  {
    "value": "1.14.2-1.amzn2.x86_64",
    "key": "parallelcluster:efa_version"
  },
  {
    "value": "alinux2",
    "key": "parallelcluster:os"
  },
  {
    "value": "aws-parallelcluster-cookbook-3.1.2",
    "key": "parallelcluster:bootstrap_file"
  },
  {
    "value": "1.8.23-10.amzn2.1.x86_64",
    "key": "parallelcluster:sudo_version"
  },
  {
    "value": "2.10.8-5.amzn2.x86_64",
    "key": "parallelcluster:lustre_version"
  }
],
"architecture": "x86_64"
},
"version": "3.1.2"
}
```

## pcluster export-cluster-logs

透過 Amazon S3 儲tar.gz存貯體將叢集的日誌匯出到本機存檔。

```
pcluster export-cluster-logs [-h]
    --bucket BUCKET_NAME
    --cluster-name CLUSTER_NAME
    [--bucket-prefix BUCKET_PREFIX]
    [--debug]
    [--end-time END_TIME]
    [--filters FILTER [FILTER ...]]
    [--keep-s3-objects KEEP_S3_OBJECTS]
    [--output-file OUTPUT_FILE]
    [--region REGION]
    [--start-time START_TIME]
```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster export-cluster-logs`。

### **--bucket *BUCKET\_NAME***

指定要將叢集日誌資料匯出至的 Amazon S3 儲存貯體的名稱。它必須與叢集位於相同的區域。

#### Note

您必須向 Amazon S3 儲存貯體政策新增許可才能授予 CloudWatch 存取權。如需詳細資訊，請參閱 CloudWatch 日誌使用者指南中的在 [Amazon S3 儲存貯體上設定許可](#)。

### **--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

### **--bucket-prefix *BUCKET\_PREFIX***

指定 Amazon S3 儲存貯體中要存放匯出日誌資料的路徑。

默認情況下，值區前綴是：

```
cluster-name-logs-202209061743.tar.gz
```

*202209061743* 是以 %Y%m%d%H%M 格式表示的目前時間。

**--debug**

啟用偵錯記錄。

**--end-time *END\_TIME***

指定收集記錄事件的時間範圍結束時間，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。不會包含時間戳記等於或晚於此時間的事件。時間元素 (例如分鐘和秒鐘) 可以省略。預設值是目前時間。

**--filters *FILTER* [*FILTER* ...]**

指定記錄的篩選器。格式:Name=a,Values=1 Name=b,Values=2,3. 支援的篩選器有：  
private-dns-name

指定執行個體私有 DNS 名稱的簡短格式 (例如 ip-10-0-0-101)。

node-type

指定節點類型，此過濾器唯一接受的值是HeadNode。

**--keep-s3-objects *KEEP\_S3\_OBJECTS***

如果 true，則會保留匯出至 Amazon S3 的匯出物件。(預設為 false。)

**--output-file *OUTPUT\_FILE***

指定儲存記錄存檔的檔案路徑。如果提供此選項，則記錄會儲存在本機。否則，它們將使用輸出中返回的 URL 上傳到 Amazon S3。預設為上傳到 Amazon S3。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

**--start-time *START\_TIME***

指定時間範圍的開始，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。包含時間戳記等於此時間或晚於此時間的記錄事件。如果未指定，預設值是建立叢集的時間。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
```

```
{  
  "url": "https://cluster-v3-bucket..."  
}
```

## pcluster export-image-logs

透過 Amazon S3 儲存貯體，將映像產生器堆疊的日誌匯出到本機tar.gz存檔。

```
pcluster export-image-logs [-h]  
    --bucket BUCKET  
    --image-id IMAGE_ID  
    [--bucket-prefix BUCKET_PREFIX]  
    [--debug]  
    [--end-time END_TIME]  
    [--keep-s3-objects KEEP_S3_OBJECTS]  
    [--output-file OUTPUT_FILE]  
    [--region REGION]  
    [--start-time START_TIME]
```

### 命名參數

#### **-h, --help**

顯示的說明文字pcluster export-image-logs。

#### **--bucket *BUCKET\_NAME***

指定要將映像建置日誌匯出到的 Amazon S3 儲存貯體名稱。它必須位於與圖像相同的區域中。

#### Note

您必須向 Amazon S3 儲存貯體政策新增許可才能授予 CloudWatch 存取權。如需詳細資訊，請參閱CloudWatch 日誌使用者指南中的在 [Amazon S3 儲存貯體上設定許可](#)。

#### **--image-id, -i *IMAGE\_ID***

將匯出其記錄檔的映像 ID。

#### **--bucket-prefix *BUCKET\_PREFIX***

指定 Amazon S3 儲存貯體中要存放匯出日誌資料的路徑。

默認情況下，值區前綴是：

```
ami-id-logs-202209061743.tar.gz
```

`202209061743`是以%Y%m%d%H%M格式表示的目前時間。

## --debug

啟用偵錯記錄。

## --end-time *END\_TIME*

指定收集記錄事件的時間範圍結束時間，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ例如 2021-01-01T20:00:00Z)。不會包含時間戳記等於或晚於此時間的事件。時間元素 (例如分鐘和秒鐘) 可以省略。預設值是目前時間。

## --keep-s3-objects *KEEP\_S3\_OBJECTS*

如果true，則會保留匯出至 Amazon S3 的匯出物件。(預設為false。)

## --output-file *OUTPUT\_FILE*

指定儲存記錄存檔的檔案路徑。如果提供此選項，則記錄會儲存在本機。否則，它們將使用輸出中返回的 URL 上傳到 Amazon S3。預設為上傳到 Amazon S3。

## --region, -r *REGION*

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

## --start-time *START\_TIME*

指定時間範圍的開始，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ例如 2021-01-01T20:00:00Z)。包含時間戳記等於此時間或晚於此時間的記錄事件。如果未指定，預設值是建立叢集的時間。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster export-image-logs --bucket image-v3-bucket --image-id ami-1234abcd5678efgh
{
  "url": "https://image-v3-bucket..."
}
```

## pcluster get-cluster-log-events

擷取與記錄資料流相關聯的事件。

```
pcluster get-cluster-log-events [-h]
    --cluster-name CLUSTER_NAME
    --log-stream-name LOG_STREAM_NAME
    [--debug]
    [--end-time END_TIME]
    [--limit LIMIT]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
    [--start-from-head START_FROM_HEAD]
    [--start-time START_TIME]
```

命名參數

**-h, --help**

顯示的說明文字pcluster get-cluster-log-events。

**--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

**--log-stream-name *LOG\_STREAM\_NAME***

指定記錄資料流的名稱。您可以使用命list-cluster-log-streams令擷取與一或多個事件相關聯的記錄資料流。

**--debug**

啟用偵錯記錄。

**--end-time *END\_TIME***

指定時間範圍的結束時間，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ例如2021-01-01T20:00:00Z)。不會包含時間戳記等於或晚於此時間的事件。

**--limit *LIMIT***

指定傳回的記錄事件數目上限。如果未指定值，最多可容納 1 MB 的記錄事件數目，最多可容納 10,000 個記錄事件。

**--next-token *NEXT\_TOKEN***

指定要用於分頁請求的權杖。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

**--start-from-head *START\_FROM\_HEAD***

如果值為true，則會先傳回最早的記錄事件。如果值為false，則會先傳回最新的記錄事件。(預設為false。)

**--start-time *START\_TIME***

指定時間範圍的開始，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ例如2021-01-01T20:00:00Z)。包含時間戳記等於此時間或晚於此時間的事件。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster get-cluster-log-events \  
  -c cluster-v3 \  
  -r us-east-1 \  
  --log-stream-name ip-198-51-100-44.i-1234567890abcdef0.clustermgtd \  
  --limit 3  
{  
  "nextToken": "f/36966906399261933213029082268132291405859205452101451780/s",  
  "prevToken": "b/36966906399239632467830551644990755687586557090595471362/s",  
  "events": [  
    {  
      "message": "2022-07-12 19:16:53,379 - [slurm_plugin.clustermgtd:_maintain_nodes]  
- INFO - Performing node maintenance actions",  
      "timestamp": "2022-07-12T19:16:53.379Z"  
    },  
    {  
      "message": "2022-07-12 19:16:53,380 - [slurm_plugin.clustermgtd:_maintain_nodes]  
- INFO - Following nodes are currently in replacement: (x0) []",  
      "timestamp": "2022-07-12T19:16:53.380Z"  
    },  
  ]  
}
```



```
{
  "message": "2022-07-12 19:16:53,380 -
[slurm_plugin.clustermgtd:_terminate_orphaned_instances] - INFO - Checking for
orphaned instance",
  "timestamp": "2022-07-12T19:16:53.380Z"
}
]
```

## pcluster get-cluster-stack-events

擷取與指定叢集的堆疊相關聯的事件。

### Note

從版本 3.6.0 開始，AWS ParallelCluster 使用巢狀堆疊來建立與佇列和計算資源相關聯的資源。GetClusterStackEventsAPI 和 pcluster get-cluster-stack-events 命令僅返回集群主堆棧事件。您可以在 CloudFormation 主控台中檢視叢集堆疊事件，包括與佇列和計算資源相關的事件。

```
pcluster get-cluster-stack-events [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

### 命名參數

#### -h, --help

顯示的說明文字 pcluster get-cluster-stack-events。

#### --cluster-name, -n *CLUSTER\_NAME*

指定叢集的名稱。

#### --debug

啟用偵錯記錄。

**--next-token** *NEXT\_TOKEN*

指定要用於分頁請求的權杖。

**--query** *QUERY*

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster get-cluster-stack-events \  
  -n cluster-v3 \  
  -r us-east-1 \  
  --query "events[0]"  
{  
  "eventId": "1234abcd-56ef-78gh-90ij-abcd1234efgh",  
  "physicalResourceId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-  
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",  
  "resourceStatus": "CREATE_COMPLETE",  
  "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-  
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",  
  "stackName": "cluster-v3",  
  "logicalResourceId": "cluster-v3",  
  "resourceType": "AWS::CloudFormation::Stack",  
  "timestamp": "2022-07-12T18:29:12.140Z"  
}
```

## **pcluster get-image-log-events**

擷取與映像版本相關聯的事件。

```
pcluster get-image-log-events [-h]  
  --image-id IMAGE_ID  
  --log-stream-name LOG_STREAM_NAME  
  [--debug]  
  [--end-time END_TIME]  
  [--limit LIMIT]  
  [--next-token NEXT_TOKEN]
```

```
[--query QUERY]  
[--region REGION]  
[--start-from-head START_FROM_HEAD]  
[--start-time START_TIME]
```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster get-image-log-events`。

### **--image-id, -i *IMAGE\_ID***

指定影像的 ID。

### **--log-stream-name *LOG\_STREAM\_NAME***

指定記錄資料流的名稱。您可以使用 `list-image-log-streams` 令擷取與一或多個事件相關聯的記錄資料流。

### **--debug**

啟用偵錯記錄。

### **--end-time *END\_TIME***

指定時間範圍的結束時間，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。不會包含時間戳記等於或晚於此時間的事件。

### **--limit *LIMIT***

指定傳回的記錄事件數目上限。如果未指定值，最多可容納 1 MB 的記錄事件數目，最多可容納 10,000 個記錄事件。

### **--next-token *NEXT\_TOKEN***

指定要用於分頁請求的權杖。

### **--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

**--start-from-head *START\_FROM\_HEAD***

如果值為true，則會先傳回最早的記錄事件。如果值為false，則會先傳回最新的記錄事件。(預設為false。)

**--start-time *START\_TIME***

指定時間範圍的開始，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。包含時間戳記等於或晚於此時間的事件。

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster get-image-log-events --image-id custom-linux2-image --region us-east-1 --  
log-stream-name 3.1.2/1 --limit 3  
{  
  "nextToken": "f/36778317771100849897800729464621464113270312017760944178/s",  
  "prevToken": "b/36778317766952911290874033560295820514557716777648586800/s",  
  "events": [  
    {  
      "message": "ExecuteBash: FINISHED EXECUTION",  
      "timestamp": "2022-04-05T22:13:26.633Z"  
    },  
    {  
      "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/  
parallelclusterimage-test-1234abcd-56ef-78gh-90ij-abcd1234efgh/3.1.2/1",  
      "timestamp": "2022-04-05T22:13:26.741Z"  
    },  
    {  
      "message": "TOE has completed execution successfully",  
      "timestamp": "2022-04-05T22:13:26.819Z"  
    }  
  ]  
}
```

**pcluster get-image-stack-events**

擷取與指定映像組建之堆疊相關聯的事件。

```
pcluster get-image-stack-events [-h]  
    --image-id IMAGE_ID  
    [--debug]  
    [--next-token NEXT_TOKEN]
```

```
[--query QUERY]
[--region REGION]
```

## 命名參數

### -h, --help

顯示的說明文字 `pcluster get-image-stack-events`。

### --image-id, -i *IMAGE\_ID*

指定影像的識別碼。

### --debug

啟用偵錯記錄。

### --next-token *NEXT\_TOKEN*

指定要用於分頁請求的權杖。

### --query *QUERY*

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster get-image-stack-events --image-id custom-linux2-image --region us-east-1 --
query "events[0]"
{
  "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
  "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
  "resourceStatus": "CREATE_IN_PROGRESS",
  "resourceStatusReason": "Resource creation Initiated",
  "resourceProperties": "{\"InfrastructureConfigurationArn\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\", \\\"ImageRecipeArn
\\\": \\\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
```

```
parallelclusterimage-custom-alinux2-image/3.1.2\", \"DistributionConfigurationArn
\": \"arn:aws:imagebuilder:us-east-1:123456789012:distribution-
configuration/parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\",
\"EnhancedImageMetadataEnabled\": \"false\", \"Tags\": {\"parallelcluster:image_name\":
\"custom-alinux2-image\", \"parallelcluster:image_id\": \"custom-alinux2-image\"}},
  \"stackId\": \"arn:aws:cloudformation:us-east-1:123456789012:stack/custom-alinux2-
image/1234abcd-56ef-78gh-90ij-abcd1234efgh\",
  \"stackName\": \"custom-alinux2-image\",
  \"logicalResourceId\": \"ParallelClusterImage\",
  \"resourceType\": \"AWS::ImageBuilder::Image\",
  \"timestamp\": \"2022-04-05T21:39:24.725Z\"
}
```

## pcluster list-clusters

擷取現有叢集的清單。

```
pcluster list-clusters [-h]
    [--cluster-status {CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
    DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
    UPDATE_COMPLETE,UPDATE_FAILED}]
    [{CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
    DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
    UPDATE_COMPLETE,UPDATE_FAILED} ...]
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

命名參數

**-h, --help**

顯示的說明文字 `pcluster list-clusters`。

**--cluster-status {CREATE\_IN\_PROGRESS, CREATE\_FAILED, CREATE\_COMPLETE, DELETE\_IN\_PROGRESS, DELETE\_FAILED, UPDATE\_IN\_PROGRESS, UPDATE\_COMPLETE, UPDATE\_FAILED} [{CREATE\_IN\_PROGRESS, CREATE\_FAILED, CREATE\_COMPLETE, DELETE\_IN\_PROGRESS, DELETE\_FAILED, UPDATE\_IN\_PROGRESS, UPDATE\_COMPLETE, UPDATE\_FAILED} ...]**

指定要篩選的叢集狀態清單。(預設為all。)

**--debug**

啟用偵錯記錄。

**--next-token *NEXT\_TOKEN***

指定要用於分頁請求的權杖。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster list-clusters
{
  "clusters": [
    {
      "clusterName": "cluster-v3",
      "cloudformationStackStatus": "CREATE_COMPLETE",
      "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
      "region": "us-east-1",
      "version": "3.1.4",
      "clusterStatus": "CREATE_COMPLETE"
    }
  ]
}
```

**pcluster list-cluster-log-streams**

擷取與叢集關聯的記錄資料流清單。

```
pcluster list-cluster-log-streams [-h]
    --cluster-name CLUSTER_NAME
    [--filters FILTERS [FILTERS ...]]
    [--next-token NEXT_TOKEN] [--debug]
    [--query QUERY]
```

```
[--region REGION]
```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster list-cluster-log-streams`。

### **--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

### **--debug**

啟用偵錯記錄。

### **--filters *FILTERS* [*FILTERS* ...]**

指定日誌串流的篩選器。格式: `Name=a,Values=1 Name=b,Values=2,3`。支援的篩選器有：

`private-dns-name`

指定執行個體私有 DNS 名稱的簡短格式 (例如 `ip-10-0-0-101`)。

`node-type`

指定節點類型，此過濾器唯一接受的值是 `HeadNode`。

### **--next-token *NEXT\_TOKEN***

指定要用於分頁請求的權杖。

### **--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster list-cluster-log-streams \  
  -n cluster-v3 \  
  -r us-east-1 \  
  --query 'logStreams[*].logStreamName'  
[
```



```
"ip-172-31-58-205.i-1234567890abcdef0.cfn-init",  
"ip-172-31-58-205.i-1234567890abcdef0.chef-client",  
"ip-172-31-58-205.i-1234567890abcdef0.cloud-init",  
"ip-172-31-58-205.i-1234567890abcdef0.clustermgtd",  
"ip-172-31-58-205.i-1234567890abcdef0.slurmctld",  
"ip-172-31-58-205.i-1234567890abcdef0.supervisord",  
"ip-172-31-58-205.i-1234567890abcdef0.system-messages"  
]
```

## pcluster list-images

擷取現有自訂映像檔的清單。

```
pcluster list-images [-h]  
    --image-status {AVAILABLE,PENDING,FAILED}  
    [--debug]  
    [--next-token NEXT_TOKEN]  
    [--query QUERY]  
    [--region REGION]
```

命名參數

### -h, --help

顯示的說明文字pcluster list-images。

### --image-status {AVAILABLE,PENDING,FAILED}

依提供的狀態篩選傳回的影像。

### --debug

啟用偵錯記錄。

### --next-token *NEXT\_TOKEN*

指定要用於分頁請求的權杖。

### --query *QUERY*

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案[default]區段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster list-images --image-status AVAILABLE
{
  "images": [
    {
      "imageId": "custom-alinux2-image",
      "imageBuildStatus": "BUILD_COMPLETE",
      "ec2AmiInfo": {
        "amiId": "ami-1234abcd5678efgh"
      },
      "region": "us-east-1",
      "version": "3.1.2"
    }
  ]
}
```

## pcluster list-image-log-streams

擷取與映像相關聯的日誌串流清單。

```
pcluster list-image-log-streams [-h]
                                --image-id IMAGE_ID
                                [--next-token NEXT_TOKEN] [--debug]
                                [--query QUERY]
                                [--region REGION]
```

命名參數

### -h, --help

顯示的說明文字 pcluster list-image-log-streams。

### --image-id, -i **IMAGE\_ID**

指定影像的識別碼。

### --debug

啟用偵錯記錄。

### --next-token **NEXT\_TOKEN**

指定要用於分頁請求的權杖。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster list-image-log-streams --image-id custom-alinux2-image --region us-east-1 --  
query 'logStreams[*].LogStreamName'  
[  
  "3.0.0/1",  
  "3.1.2/1"  
]
```

## **pcluster list-official-images**

描述官方 AWS ParallelCluster AMI。

```
pcluster list-official-images [-h]  
    [--architecture ARCHITECTURE]  
    [--debug]  
    [--os OS]  
    [--query QUERY]  
    [--region REGION]
```

### 命名參數

**-h, --help**

顯示的說明文字 `pcluster list-official-images`。

**--architecture *ARCHITECTURE***

指定用來篩選結果的架構。如果未指定此參數，則會傳回所有架構。

**--debug**

啟用偵錯記錄。

**--os *OS***

指定用來篩選結果的作業系統。如果未指定此參數，則會傳回所有作業系統。

**--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用影像組態檔案中的「區域」([Region](#)) 設定、AWS\_DEFAULT\_REGION環境變數、~/.aws/config檔案區[default]段中的region設定或--region參數來指定。

使用 AWS ParallelCluster 版本 3.1.2 的示例：

```
$ pcluster list-official-images
{
  "images": [
    {
      "amiId": "ami-015cfefb4e0d6306b2",
      "os": "ubuntu2004",
      "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-x86_64-202202261505
2022-02-26T15-08-34.759Z",
      "version": "3.1.2",
      "architecture": "x86_64"
    },
    {
      "amiId": "ami-036f23237ce49d25b",
      "os": "ubuntu2204",
      "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-x86_64-202202261505
2022-02-26T15-08-17.558Z",
      "version": "3.1.2",
      "architecture": "x86_64"
    },
    {
      "amiId": "ami-09e5327e694d89ef4",
      "os": "ubuntu2004",
      "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.736Z",
      "version": "3.1.2",
      "architecture": "arm64"
    },
    {
```

```

    "amiId": "ami-0b9b0874c35f626ae",
    "os": "alinux2",
    "name": "aws-parallelcluster-3.1.2-amzn2-hvm-x86_64-202202261505
2022-02-26T15-08-31.311Z",
    "version": "3.1.2",
    "architecture": "x86_64"
  },
  {
    "amiId": "ami-0bf6d01f398f3737e",
    "os": "centos7",
    "name": "aws-parallelcluster-3.1.2-centos7-hvm-x86_64-202202261505
2022-02-26T15-08-25.001Z",
    "version": "3.1.2",
    "architecture": "x86_64"
  },
  {
    "amiId": "ami-0d0de4f95f56374bc",
    "os": "alinux2",
    "name": "aws-parallelcluster-3.1.2-amzn2-hvm-arm64-202202261505
2022-02-26T15-08-46.088Z",
    "version": "3.1.2",
    "architecture": "arm64"
  },
  {
    "amiId": "ami-0ebf7bc54b8740dc6",
    "os": "ubuntu2204",
    "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.293Z",
    "version": "3.1.2",
    "architecture": "arm64"
  }
]
}

```

## pcluster ssh

執行具有預先填入叢集使用者名稱和 IP 位址的ssh命令。任意引數會附加到ssh命令列的結尾。

```

pcluster ssh [-h]
              --cluster-name CLUSTER_NAME
              [--debug]
              [--dryrun DRYRUN]
              [--region REGION]

```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster ssh`。

### **--cluster-name, -n *CLUSTER\_NAME***

指定要連接的叢集名稱。

### **--debug**

啟用偵錯記錄。

### **--dryrun *DRYRUN***

何時 `true`，打印將運行並退出的命令行。(預設為 `false`。)

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

範例：

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

使用預先填入叢集的使用者名稱和 IP 位址執行 `ssh` 命令：

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

## **pcluster update-cluster**

更新現有叢集以符合指定組態檔的設定。

```
pcluster update-cluster [-h]
                        --cluster-configuration CLUSTER_CONFIGURATION
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--dryrun DRYRUN]
                        [--force-update FORCE_UPDATE]
                        [--query QUERY]
                        [--region REGION]
                        [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
```

```
[--validation-failure-level {INFO,WARNING,ERROR}]
```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster update-cluster`。

### **--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

指定 YAML 叢集配置檔案。

### **--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

### **--debug**

啟用偵錯記錄。

### **--dryrun *DRYRUN***

何時 `true`，執行驗證而不更新叢集和建立任何資源。它可以用來驗證映像配置和更新需求。(預設為 `false`。)

### **--force-update *FORCE\_UPDATE***

何時 `true`，會忽略更新驗證錯誤來強制更新。(預設為 `false`。)

### **--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用叢集配置檔案中的 [Region](#) 設定、`AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

識別要抑制的一或多個組態驗證程式。

格式: `(ALL|type:[A-Za-z0-9]+)`

### **--validation-failure-level *{INFO,WARNING,ERROR}***

指定針對更新報告的驗證失敗層級。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster update-cluster -c cluster-config.yaml -n cluster-v3 -r us-east-1
{
  "cluster": {
    "clusterName": "cluster-v3",
    "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
    "region": "us-east-1",
    "version": "3.1.4",
    "clusterStatus": "UPDATE_IN_PROGRESS"
  },
  "changeSet": [
    {
      "parameter": "HeadNode.Iam.S3Access",
      "requestedValue": {
        "BucketName": "pc-beta-test",
        "KeyName": "output",
        "EnableWriteAccess": false
      }
    },
    {
      "parameter": "HeadNode.Iam.S3Access",
      "currentValue": {
        "BucketName": "pcluster-east-test-bucket",
        "KeyName": "logs",
        "EnableWriteAccess": true
      }
    }
  ]
}
```

## pcluster update-compute-fleet

更新叢集運算叢集的狀態。

```
pcluster update-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    --status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

    [--debug]
    [--query QUERY]
```



```
[--region REGION]
```

## 命名參數

### **-h, --help**

顯示的說明文字 `pcluster update-compute-fleet`。

### **--cluster-name, -n *CLUSTER\_NAME***

指定叢集的名稱。

### **--status {*START\_REQUESTED, STOP\_REQUESTED, ENABLED, DISABLED*}**

指定套用至叢集運算叢集的狀態。狀態 `START_REQUESTED` 和 `STOP_REQUESTED` 對應於 Slurm 調度程序，而狀態 `ENABLED` 和 `DISABLED` 對應於 AWS Batch 調度程序。

### **--debug**

啟用偵錯記錄。

### **--query *QUERY***

指定要在輸出上執行的 JMESPath 查詢。

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

指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster update-compute-fleet -n cluster-v3 --status STOP_REQUESTED
{
  "status": "STOP_REQUESTED",
  "lastStatusUpdateTime": "2022-07-12T20:19:47.653Z"
}
```

## **pcluster version**

顯示的版本 AWS ParallelCluster。

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

## 命名參數

### -h, --help

顯示的說明文字pcluster version。

### --debug

啟用偵錯記錄。

使用 AWS ParallelCluster 版本 3.1.4 的示例：

```
$ pcluster version
{
  "version": "3.1.4"
}
```

## pcluster3-config-converter

讀取AWS ParallelCluster版本 2 配置檔案並寫入AWS ParallelCluster版本 3 配置檔案。

```
pcluster3-config-converter [-h]
                          [-t CLUSTER_TEMPLATE]
                          [-c CONFIG_FILE]
                          [--force-convert]
                          [-o OUTPUT_FILE]
```

## 命名參數

### -h, --help

顯示的說明文字pcluster3-config-converter。

### -t **CLUSTER\_TEMPLATE**, --cluster-template **CLUSTER\_TEMPLATE**

指定要轉換的[\[cluster\]](#)組態檔案區段。如果未指定，指令碼會在[\[global\]](#)區段中尋找[叢集範本](#)參數，或將搜尋。[\[cluster default\]](#)

### -c **CONFIG\_FILE**, --config-file **CONFIG\_FILE**

指定要讀取的AWS ParallelCluster版本 2 組態檔。

## --force-convert

即使不支援一個或多個設定，也可以啟用轉換。

## -o *OUTPUT\_FILE*, --output-file *OUTPUT\_FILE*

指定要寫入的AWS ParallelCluster版本 3 規劃檔。如果未指定此參數，則會將配置寫入 stdout。

### Note

在AWS ParallelCluster版本 3.0.1 中添加了該pcluster3-config-converter命令。

## 組態檔案

AWS ParallelCluster 使用 YAML 1.1 檔案做為組態參數。

### 主題

- [叢集配置檔](#)
- [構建映像配置文件](#)

## 叢集配置檔

AWS ParallelCluster 第 3 版使用不同的組態檔來控制叢集基礎結構的定義和自訂 AMI 的定義。所有配置文件都使用 YAML 1.1 文件。每個組態檔案的詳細資訊連結如下。如需一些設定範例，請參閱 [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example\\_configs](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs)。

這些物件用於 AWS ParallelCluster 版本 3 叢集配置。

### 主題

- [叢集配置檔屬性](#)
- [Imds 區段](#)
- [Image 區段](#)
- [HeadNode 區段](#)
- [Scheduling 區段](#)
- [SharedStorage 區段](#)
- [iam 區段](#)

- [LoginNodes 區段](#)
- [Monitoring 區段](#)
- [Tags 區段](#)
- [AdditionalPackages 區段](#)
- [DirectoryService 區段](#)
- [DeploymentSettings 區段](#)

## 叢集配置檔屬性

Region ( 可選 , String )

指定 AWS 區域 叢集的。例如 us-east-2。

更新原則：如果變更此設定，則不允許更新。

CustomS3Bucket ( 可選 , String )

指定在您用 AWS 帳戶 來存放叢集使用的資源 (例如叢集組態檔) 中建立的 Amazon S3 儲存貯體的名稱。AWS ParallelCluster 在您建立叢集的每個 AWS 區域 儲存貯體中維護一個 Amazon S3 儲存貯體。根據預設，這些 Amazon S3 儲存貯體會被命名為parallelcluster-*hash*-v1-DO-NOT-DELETE。

更新原則：如果變更此設定，則不允許更新。如果強制更新，則會忽略新值，並使用舊值。

AdditionalResources ( 可選 , String )

定義要隨叢集一起啟動的其他 AWS CloudFormation 範本。這個額外的範本可用來建立叢集外部但屬於叢集生命週期的一部分的資源。

該值必須是公用範本的 HTTPS URL，並提供了所有參數。

沒有預設值。

更新原則：您可以在更新期間變更此設定。

## Imds 區段

(選擇性) 指定全域執行個體中繼資料服務 (IMDS) 組態。

[Imds](#) :

`ImdsSupport`: *string*

## Imds 屬性

`ImdsSupport` ( 可選 , String )

指定叢集節點支援哪些 IMDS 版本。支援的值為 `v1.0` 和 `v2.0`。預設值為 `v2.0`。

如果設定 `ImdsSupport` 為 `v1.0`，則同時支援 `IMDSv1` 和 `IMDSv2`。

如果設定 `ImdsSupport` 為 `v2.0`，則僅支援 `IMDSv2`。

如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的使用 [IMDSv2](#)。

更新原則：如果變更此設定，則不允許更新。

### Note

從 AWS ParallelCluster 3.7.0 開始，`ImdsSupport` 預設值為 `v2.0`。建議您在自訂動作呼叫中設定 `ImdsSupport` 為 `IMDSv1`，`v2.0` 並將其取代為 `IMDSv2`。

Support [Imds/ImdsSupport](#) 添加了 3.3.0 AWS ParallelCluster 版本。

## Image 區段

(必要) 定義叢集的作業系統。

`Image`:

`Os`: *string*

`CustomAmi`: *string*

## Image 屬性

`Os` ( 必填 , String )

指定要用於叢集的作業系統。支援的值

為 `alinux2`、`alinux2023`、`centos7`、`ubuntu2204`、`ubuntu2004`、`rhel8`、`rocky8`、`rhel9`、`r`

### Note

RedHat 從 3.6.0 版開始，已新增企業 AWS ParallelCluster 版 Linux 8.7 (`rhel8`)。

如果您將叢集設定為使用 `rhel`，則任何執行個體類型的隨需費用都會高於將叢集設定為使用其他支援的作業系統時的成本。如需有關定價的詳細資訊，請參閱 [隨需定價](#) 以及 [Amazon EC2 上的 RHEL 是如何提供和定價的？](#)。

RedHat 從 3.9.0 版本開始增加了企業 AWS ParallelCluster 版本 9 (RHEL9)。

除了下表中 AWS 區域 提到的不支持的具體內容 `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)                                       | <code>alinux2</code> | <code>centos7</code> | <code>ubuntu2204</code> 和 <code>ubuntu2004</code> | <code>rhel8</code> | <code>rhel9</code> | <code>alinux2023</code> |
|----------------------------------------------------|----------------------|----------------------|---------------------------------------------------|--------------------|--------------------|-------------------------|
| 商業 (所有 AWS 區域 未具體提及)                               | True                 | True                 | True                                              | True               | True               | True                    |
| AWS GovCloud (美國東部) ( <code>us-gov-east-1</code> ) | True                 | False                | True                                              | True               | True               | True                    |
| AWS GovCloud (美國西部) ( <code>us-gov-west-1</code> ) | True                 | False                | True                                              | True               | True               | True                    |
| 中國 (北京) ( <code>cn-north-1</code> )                | True                 | False                | True                                              | True               | True               | True                    |
| 中國 (寧夏) ( <code>cn-northwest-1</code> )            | True                 | False                | True                                              | True               | True               | True                    |

更新原則：如果變更此設定，則不允許更新。

#### Note

AWS ParallelCluster 3.8.0 支持洛基 Linux 8，但預建的落基 Linux 8 AMI (對於 x86 和 ARM 架構) 是不可用的。AWS ParallelCluster 3.8.0 支援使用自訂 AMI 使用岩石 Linux 8 建立叢集。如需詳細資訊，請參閱 [作業系統考量](#)。AWS ParallelCluster 3.9.0 支持洛

基 Linux 9，但預建的落基 Linux 9 AMI (對於 x86 和 ARM 架構) 是不可用的。AWS ParallelCluster 3.9.0 支援使用自訂 AMI 使用岩石 Linux 9 建立叢集。如需詳細資訊，請參閱 [作業系統考量](#)。

### CustomAmi (可選, String)

指定要用於頭節點和計算節點的自訂 AMI 的識別碼，而非預設 AMI。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 定制](#)。

如果自訂 AMI 需要其他權限才能啟動，則必須將這些權限新增至使用者和頭節點原則。

例如，如果自訂 AMI 具有與其相關聯的加密快照，則使用者和標頭節點原則都需要下列其他原則：

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

若要建置 RedHat 企業 Linux 自訂 AMI，您必須設定作業系統以安裝 RHUI (AWS) 儲存庫所提供的套件：`rhel-<version>-baseos-rhui-rpms`、`rhel-<version>-appstream-rhui-rpms`、`codeready-builder-for-rhel-<version>-rhui-rpms`。此外，自定義 AMI 上的存儲庫必須包含 `kernel-devel` 包含與正在運行的內核版本相同版本的軟件包。kernel。

已知限制：

- 只有 RHEL 8.2 及更新版本支援 FSx for Lustre。
- RHEL 8.7 核心版本 4.18.0-425.3.1.1.el8 不支援 FSx for Lustre。

- 只有 RHEL 8.4 及更新版本支援 EFA。
- AL23 不支援 NICE DCV，因為它不包含執行 NICE DCV 所需的圖形桌面環境。如需詳細資訊，請參閱官方 [NICE DCV 文件](#)。

若要疑難排解自訂 AMI 驗證警告，請參閱[排解自訂 AMI 問題](#)。

更新原則：如果變更此設定，則不允許更新。

## HeadNode 區段

(必要) 指定頭節點的組態。

```
HeadNode:
  InstanceType: string
  Networking:
    SubnetId: string
    ElasticIp: string/boolean
    SecurityGroups:
      - string
    AdditionalSecurityGroups:
      - string
  Proxy:
    HttpProxyAddress: string
  DisableSimultaneousMultithreading: boolean
  Ssh:
    KeyName: string
    AllowedIps: string
  LocalStorage:
    RootVolume:
      Size: integer
      Encrypted: boolean
      VolumeType: string
      Iops: integer
      Throughput: integer
      DeleteOnTermination: boolean
    EphemeralVolume:
      MountDir: string
  SharedStorageType: string
  Dcv:
    Enabled: boolean
    Port: integer
    AllowedIps: string
```



```
CustomActions:  
  OnNodeStart:  
    Sequence:  
      - Script: string  
        Args:  
          - string  
        Script: string  
        Args:  
          - string  
    OnNodeConfigured:  
      Sequence:  
        - Script: string  
          Args:  
            - string  
        Script: string  
        Args:  
          - string  
    OnNodeUpdated:  
      Sequence:  
        - Script: string  
          Args:  
            - string  
        Script: string  
        Args:  
          - string  
Iam:  
  InstanceRole: string  
  InstanceProfile: string  
  S3Access:  
    - BucketName: string  
      EnableWriteAccess: boolean  
      KeyName: string  
  AdditionalIamPolicies:  
    - Policy: string  
Imds:  
  Secured: boolean  
Image:  
  CustomAmi: string
```

## HeadNode 屬性

InstanceType ( 必填 , String )

指定頭節點的執行個體類型。

指定用於標頭節點的 Amazon EC2 執行個體類型。執行個體類型的架構必須與 AWS Batch [InstanceType](#) 或 Slurm [InstanceType](#) 設定所使用的架構相同。

**Note**

AWS ParallelCluster 設定不支援下列執行個體類 HeadNode 型。

- hpc6id

如果您定義 p4d 執行個體類型或具有多個網路介面卡或網路介面卡的其他執行個體類型，則必須 [ElasticIp](#) 將設定為提供公用存取權。AWS 只能將公有 IP 指派給透過單一網路介面啟動的執行個體。在此情況下，建議您使用 [NAT 閘道](#) 來提供叢集運算節點的公用存取權。[如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公用 IPv4 地址。](#)

更新原則：如果變更此設定，則不允許更新。

`DisableSimultaneousMultithreading` (可選, Boolean)

如果 `true`，停用頭節點上的超執行緒。預設值為 `false`。

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](#)。

更新原則：如果變更此設定，則不允許更新。

`SharedStorageType` (可選, String)

指定用於內部共用資料的儲存體類型。內部共用資料包括 AWS ParallelCluster 用來管理叢集的資料，`/home` 如果未在中指定 [SharedStorage](#) 區段為掛載目錄來掛載共用檔案系統磁碟區，則預設為共用資料。如需內部共用資料的詳細資訊，請參閱 [AWS ParallelCluster 內部目錄](#)。

如果 `Ebs` 是預設儲存類型，則頭節點會將其根磁碟區的一部分匯出為使用 NFS 的計算節點和登入節點的共用目錄。

如果 `Efs`，平行叢集將建立 EFS 檔案系統，以用於共用內部資料和 `/home`

更新原則：如果變更此設定，則不允許更新。

**Note**

當叢集向外延展時，EBS 儲存區類型可能會出現效能瓶頸，因為頭節點與使用 NFS 匯出的運算節點共用根磁碟區的資料。使用 EFS，您可以在叢集向外擴充時避免 NFS 匯出，並避

免與它們相關的效能瓶頸。建議您選擇 EBS，以獲得小型檔案和安裝程序的最大讀取/寫入潛力。選擇 EFS 做為縮放。

## Networking

(必要) 定義頭節點的網路組態。

### Networking:

SubnetId: *string*

ElasticIp: *string/boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

Proxy:

HttpProxyAddress: *string*

更新原則：如果變更此設定，則不允許更新。

## Networking 屬性

SubnetId ( 必填 , String )

指定要在其中佈建標頭節點的現有子網路識別碼。

更新原則：如果變更此設定，則不允許更新。

ElasticIp ( 可選 , String )

建立或指派彈性 IP 位址給頭節點。支援的值為truefalse、或現有彈性 IP 位址的識別碼。預設值為 false。

更新原則：如果變更此設定，則不允許更新。

SecurityGroups ( 可選 , [String] )

要用於頭節點的 Amazon VPC 安全群組識別碼清單。這些會取代如果未包含此內容時所 AWS ParallelCluster 建立的安全性群組。

確認您的 [SharedStorage](#) 系統已正確設定安全群組。

更新原則：您可以在更新期間變更此設定。

## AdditionalSecurityGroups ( 可選 , [String] )

要用於頭節點的其他 Amazon VPC 安全群組識別碼清單。

[更新原則：您可以在更新期間變更此設定。](#)

## Proxy(選擇性)

指定頭節點的代理伺服器設定。

```
Proxy:  
  HttpProxyAddress: string
```

## HttpProxyAddress ( 可選 , String )

定義 HTTP 或 HTTPS 代理伺服器，通常為 `https://x.x.x.x:8080`。

沒有預設值。

[更新原則：如果變更此設定，則不允許更新。](#)

## Ssh

(選擇性) 定義 SSH 存取頭節點的組態。

```
Ssh:  
  KeyName: string  
  AllowedIps: string
```

[更新原則：您可以在更新期間變更此設定。](#)

## Ssh 屬性

### KeyName ( 可選 , String )

命名現有的 Amazon EC2 key pair，以啟用對頭節點的 SSH 存取。

[更新原則：如果變更此設定，則不允許更新。](#)

### AllowedIps ( 可選 , String )

指定與頭節點之 SSH 連線的 CIDR 格式 IP 範圍或首碼清單識別碼。預設值為 `0.0.0.0/0`。

[更新原則：您可以在更新期間變更此設定。](#)

## LocalStorage

(選擇性) 定義頭節點的本機儲存配置。

```
LocalStorage:  
  RootVolume:  
    Size: integer  
    Encrypted: boolean  
    VolumeType: string  
    Iops: integer  
    Throughput: integer  
    DeleteOnTermination: boolean  
  EphemeralVolume:  
    MountDir: string
```

更新原則：您可以在更新期間變更此設定。

### LocalStorage 屬性

#### RootVolume(必填)

指定頭節點的根磁碟區儲存體。

```
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  VolumeType: string  
  Iops: integer  
  Throughput: integer  
  DeleteOnTermination: boolean
```

更新原則：您可以在更新期間變更此設定。

#### Size ( 可選 , Integer )

以 gib (GiB) 為單位指定頭節點根磁碟區的大小。預設大小來自 AMI。使用不同的大小需要 AMI 支援growroot。

更新原則：如果變更此設定，則不允許更新。

#### Encrypted ( 可選 , Boolean )

指定根磁碟區是否已加密。預設值為 true。

更新原則：如果變更此設定，則不允許更新。

VolumeType ( 可選 , String )

指定 [Amazon EBS 磁碟區類型](#)。支援的值為gp2gp3、io1、io2、sc1、st1、和standard。預設值為 gp3。

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](#)。

更新原則：如果變更此設定，則不允許更新。

Iops ( 可選 , Integer )

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VolumeType不同Size。

更新原則：如果變更此設定，則不允許更新。

VolumeType = io1

預設值 Iops =

支援的值 Iops =

最大Iops比Size率 = 每 GiB 50 IOPS。5000 IOPS 需要一Size個至少 100 GiB。

VolumeType = io2

預設值 Iops =

支援的值 Iops = 100 至 64 萬 (對於io2區塊快速磁碟區為 256000) †

最大Iops比Size率 = 每 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB 博Size的一個。

VolumeType = gp3

預設Iops值

支援的Iops值

最大Iops比Size率 = 每 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB 博Size的一個。

† 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 區塊快速磁碟區在 R5b 執行個體類型上支援高達 256000 的 Iops 值。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](#)。

更新原則：您可以在更新期間變更此設定。

Throughput ( 可選 , Integer )

定義 gp3 磁碟區類型的輸送量，以 MIB/s 為單位。此設定只有在為時 VolumeType 才有效 gp3。預設值為 125。支援的值：每秒

與的比率 Iops 可 Throughput 以不超過 0.25。每秒 1000 MB 的最大輸送量需要 Iops 設定至少為 4000。

更新原則：如果變更此設定，則不允許更新。

DeleteOnTermination ( 可選 , Boolean )

指定當頭節點終止時，是否應刪除根磁碟區。預設值為 true。

更新原則：如果變更此設定，則不允許更新。

EphemeralVolume(選擇性)

指定任何執行個體儲存量的詳細資料。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [《執行個體儲存體磁碟區》](#)。

```
EphemeralVolume:  
  MountDir: string
```

更新原則：如果變更此設定，則不允許更新。

MountDir ( 可選 , String )

指定執行個體儲存磁碟區的掛載目錄。預設值為 /scratch。

更新原則：如果變更此設定，則不允許更新。

## Dcv

(選擇性) 為在頭節點上執行的 NICE DCV 伺服器定義組態設定。

如需詳細資訊，請參閱 [通過 NICE DCV Connect 到頭節點](#)。

```
Dcv:
```

`Enabled`: *boolean*  
`Port`: *integer*  
`AllowedIps`: *string*

### Important

根據預設，NICE DCV 連接埠設定方式 AWS ParallelCluster 會開放給所有 IPv4 位址。但是，只有當您擁有 NICE DCV 工作階段的 URL，並在傳回 URL 後的 30 秒內連線到 NICE DCV 工作階段時，才可以連線到 NICE DCV 連接埠。pcluster dcv-connect 使用此 AllowedIps 設定可進一步限制對具有 CIDR 格式 IP 範圍之 NICE DCV 連接埠的存取，並使用此 Port 設定來設定非標準連接埠。

更新原則：如果變更此設定，則不允許更新。

## Dcv 屬性

Enabled ( 必填 , Boolean )

指定是否在頭節點上啟用 NICE DCV。預設值為 false。

更新原則：如果變更此設定，則不允許更新。

### Note

NICE DCV 會自動生成一個自簽名證書，用於保護在頭節點上運行的 NICE DCV 客戶端和 NICE DCV 服務器之間的流量。若要設定您自己的憑證，請參閱 [NICE DCV DCV HTTPS 證書](#)。

Port ( 可選 , Integer )

指定 NICE DCV DCV 的連接埠。預設值為 8443。

更新原則：如果變更此設定，則不允許更新。

AllowedIps ( 可選 , 推薦 , String )

為連線至 NICE DCV 指定 CID 格式的 IP 範圍。只有在建立安全性群組時才 AWS ParallelCluster 會使用此設定。預設值是 0.0.0.0/0，該值允許從任何網際網路位址存取。



更新原則：您可以在更新期間變更此設定。

## CustomActions

(選擇性) 指定要在頭節點上執行的自訂指令集。

```
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeConfigured:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeUpdated:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
```

### CustomActions 屬性

#### OnNodeStart(選擇性)

指定在啟動任何節點部署啟動程序動作之前，要在頭節點上執行的單一指令碼或一系列指令碼。如需詳細資訊，請參閱 [自訂引導動作](#)。

#### Sequence(選擇性)

要執行的指令碼清單。AWS ParallelCluster 以與組態檔案中列出的順序相同，從第一個開始執行指令碼。

Script ( 必填 , String )

指定要使用的檔案。檔案路徑可以以https://或開頭s3://。

Args ( 可選 , [String] )

要傳遞至指令碼的引數清單。

Script ( 必填 , String )

指定要用於單一腳本的檔案。檔案路徑可以以https://或開頭s3://。

Args ( 可選 , [String] )

要傳遞給單一指令碼的引數清單。

更新原則：如果變更此設定，則不允許更新。

OnNodeConfigured(選擇性)

指定節點啟動程序動作完成後，要在頭節點上執行的單一指令碼或一系列指令碼。如需詳細資訊，請參閱 [自訂引導動作](#)。

Sequence(選擇性)

指定要執行的指令碼清單。

Script ( 必填 , String )

指定要使用的檔案。檔案路徑可以以https://或開頭s3://。

Args ( 可選 , [String] )

要傳遞至指令碼的引數清單。

Script ( 必填 , String )

指定要用於單一腳本的檔案。檔案路徑可以以https://或開頭s3://。

Args ( 可選 , [String] )

要傳遞給單一指令碼的引數清單。

更新原則：如果變更此設定，則不允許更新。

OnNodeUpdated(選擇性)

指定節點更新動作完成後，要在頭節點上執行的單一指令碼或一系列指令碼。如需詳細資訊，請參閱 [自訂引導動作](#)。

## Sequence(選擇性)

指定要執行的指令碼清單。

Script ( 必填 , String )

指定要使用的檔案。檔案路徑可以以https://或開頭s3://。

Args ( 可選 , [String] )

要傳遞至指令碼的引數清單。

Script ( 必填 , String )

指定要用於單一腳本的檔案。檔案路徑可以以https://或開頭s3://。

Args ( 可選 , [String] )

要傳遞給單一指令碼的引數清單。

[更新原則：您可以在更新期間變更此設定。](#)

### Note

OnNodeUpdated已從 AWS ParallelCluster 3.4.0 開始新增。

Sequence從 AWS ParallelCluster 版本 3.6.0 開始添加。當您指定時Sequence，您可以列出自訂動作的多個指令碼。AWS ParallelCluster 繼續支援使用單一指令碼設定自訂動作，但不包含Sequence。

AWS ParallelCluster 不支持包括單個腳本和相同Sequence的自定義操作。

## Iam

(選擇性) 指定要在標頭節點上使用的執行個體角色或執行個體設定檔，以覆寫叢集的預設執行個體角色或執行個體設定檔。

Iam:

InstanceRole: *string*

InstanceProfile: *string*

S3Access:

- BucketName: *string*

EnableWriteAccess: *boolean*

KeyName: *string*

AdditionalIamPolicies:

- `Policy`: *string*

更新原則：您可以在更新期間變更此設定。

## Iam 屬性

InstanceProfile ( 可選 , String )

指定例證紀要以取代預設標頭節點例證輪廓。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`。

如果已指定，則無法指AdditionalIamPolicies定S3Access和設定。

建議您指定其中一個或兩個S3Access和AdditionalIamPolicies設定，因為新增至的功能 AWS ParallelCluster 通常需要新的權限。

更新原則：如果變更此設定，則不允許更新。

InstanceRole ( 可選 , String )

指定要覆寫預設標頭節點執行個體角色的執行個體角色。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:role/RoleName`。

如果已指定，則無法指AdditionalIamPolicies定S3Access和設定。

建議您指定其中一個或兩個S3Access和AdditionalIamPolicies設定，因為新增至的功能 AWS ParallelCluster 通常需要新的權限。

更新原則：您可以在更新期間變更此設定。

## S3Access

S3Access(選擇性)

指定值區。這是用來產生政策，以授與值區的指定存取權。

如果已指定，則無法指InstanceRole定InstanceProfile和設定。

建議您指定其中一個或兩個S3Access和AdditionalIamPolicies設定，因為新增至的功能 AWS ParallelCluster 通常需要新的權限。

S3Access:

```
- BucketName: string  
EnableWriteAccess: boolean  
KeyName: string
```

更新原則：您可以在更新期間變更此設定。

BucketName ( 必填 , String )

儲存貯體的名稱。

更新原則：您可以在更新期間變更此設定。

KeyName ( 可選 , String )

存儲桶的關鍵。預設值為 \*。

更新原則：您可以在更新期間變更此設定。

EnableWriteAccess ( 可選 , Boolean )

指出是否已啟用值區的寫入存取權。預設值為 false。

更新原則：您可以在更新期間變更此設定。

## AdditionalIamPolicies

AdditionalIamPolicies(選擇性)

為 Amazon Amazon EC2 指定 IAM 政策的亞馬遜資源名稱 (ARN) 清單。除了所需的權限之外，此清單還附加至用於頭節點的根角色 AWS ParallelCluster。

IAM 政策名稱及其 ARN 不同。不能使用名稱。

如果已指定，則無法指 InstanceRole 定 InstanceProfile 和設定。

我們建議您使用，AdditionalIamPolicies 因 AdditionalIamPolicies 為已新增至 AWS ParallelCluster 需要的權限，且 InstanceRole 必須包含所有必要的權限。所需的許可經常隨發行版本新增功能而變更。

沒有預設值。

```
AdditionalIamPolicies:  
- Policy: string
```

更新原則：您可以在更新期間變更此設定。

Policy ( 可選 , [String] )

IAM 政策清單。

更新原則：您可以在更新期間變更此設定。

## Imds

(選擇性) 指定執行個體中繼資料服務 (IMDS) 的屬性。 [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服務第 2 版的運作方式。](#)

Imds:

Secured: *boolean*

更新原則：如果變更此設定，則不允許更新。

## Imds 屬性

Secured ( 可選 , Boolean )

如果true，則限制對頭節點的 IMDS ( 以及執行個體設定檔認證 ) 的存取權限為超級使用者的子集。

如果false，頭節點中的每個用戶都可以訪問頭節點的 IMDS。

允許下列使用者存取頭節點的 IMDS：

- 根使用者
- 叢集管理使用pc-cluster-admin者 (預設)
- 特定於操作系統的默認用戶 ( ec2-user在 Amazon Linux 2 和 Ubuntu 18.04 ubuntu 上 RedHat , centos在 CentOS 7 上 )

預設值為 true。

default使用者必須負責確保叢集具有與 AWS 資源互動所需的權限。如果停用default使用者 IMDS 存取權，則 AWS ParallelCluster 無法管理運算節點並停止運作。請勿停用default使用者 IMDS 存取權。

當用戶被授予對頭節點 IMDS 的訪問權限時，他們可以使用[頭節點的實例配置文件中包含的](#)權限。例如，他們可以使用這些許可來啟動 Amazon EC2 執行個體，或讀取叢集設定為用於身份驗證之 AD 網域的密碼。

若要限制 IMDS 存取，請 AWS ParallelCluster 管理的 iptables 鏈結。

具有 sudo 存取權的叢集使用者可以透過執行以下命令，選擇性地啟用或停用其他個別使用者 (包括使用 default 者) 對頭節點之 IMDS 的存取：

```
$ sudo /opt/parallelcluster/scripts/imds/imds-access.sh --allow <USERNAME>
```

您可以使用此命令的 --deny 選項來停用使用者 IMDS 存取。

如果您在不知不覺中停用使用 default 者 IMDS 存取權限，您可以使用選項還原權限。--allow

#### Note

iptables 或 ip6tables 規則的任何自訂都可能干擾用來限制頭節點上 IMDS 存取的機制。

更新原則：如果變更此設定，則不允許更新。

## Image

(選擇性) 定義頭節點的自訂影像。

```
Image:  
  CustomAmi: string
```

更新原則：如果變更此設定，則不允許更新。

## Image 屬性

CustomAmi (可選, String)

指定要用於頭節點而非預設 AMI 的自訂 AMI 識別碼。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 定制](#)。

如果自訂 AMI 需要其他權限才能啟動，則必須將這些權限新增至使用者和頭節點原則。

例如，如果自訂 AMI 具有與其相關聯的加密快照，則使用者和標頭節點原則都需要下列其他原則：

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "ec2:DescribeSnapshots",  
      "Resource": "*" } ] }
```

```

    {
      "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>"
      ]
    }
  ]
}

```

若要疑難排解自訂 AMI 驗證警告，請參閱[排解自訂 AMI 問題](#)。

更新原則：如果變更此設定，則不允許更新。

## Scheduling 區段

(必要) 定義叢集中使用的工作排程器，以及工作排程器管理的計算執行個體。您可以使用Slurm或AWS Batch 排程器。每個都支援一組不同的設定和屬性。

### 主題

- [Scheduling 屬性](#)
- [AwsBatchQueues](#)
- [SlurmQueues](#)
- [SlurmSettings](#)

#### Scheduling:

Scheduler: slurm

ScalingStrategy: *string*

#### SlurmSettings:

MungeKeySecretArn: *string*

ScaledownIdleTime: *integer*

QueueUpdateStrategy: *string*

EnableMemoryBasedScheduling: *boolean*

CustomSlurmSettings: [*dict*]

CustomSlurmSettingsIncludeFile: *string*



Database:Uri: *string*UserName: *string*PasswordSecretArn: *string*DatabaseName: *string*ExternalSlurmdbd: *boolean*Host: *string*Port: *integer*Dns:DisableManagedDns: *boolean*HostedZoneId: *string*UseEc2Hostnames: *boolean*SlurmQueues:

- Name: *string*

ComputeSettings:LocalStorage:RootVolume:Size: *integer*Encrypted: *boolean*VolumeType: *string*Iops: *integer*Throughput: *integer*EphemeralVolume:MountDir: *string*CapacityReservationTarget:CapacityReservationId: *string*CapacityReservationResourceGroupArn: *string*CapacityType: *string*AllocationStrategy: *string*JobExclusiveAllocation: *boolean*CustomSlurmSettings: *dict*Tags:

- Key: *string*

Value: *string*HealthChecks:Gpu:Enabled: *boolean*Networking:SubnetIds:

- *string*

AssignPublicIp: *boolean*SecurityGroups:

- *string*

AdditionalSecurityGroups:

```
- string
PlacementGroup:
  Enabled: boolean
  Id: string
  Name: string
Proxy:
  HttpProxyAddress: string
ComputeResources:
- Name: string
  InstanceType: string
  Instances:
    - InstanceType: string
  MinCount: integer
  MaxCount: integer
  DynamicNodePriority: integer
  StaticNodePriority: integer
  SpotPrice: float
  DisableSimultaneousMultithreading: boolean
  SchedulableMemory: integer
  HealthChecks:
    Gpu:
      Enabled: boolean
  Efa:
    Enabled: boolean
    GdrSupport: boolean
  CapacityReservationTarget:
    CapacityReservationId: string
    CapacityReservationResourceGroupArn: string
  Networking:
    PlacementGroup:
      Enabled: boolean
      Name: string
    CustomSlurmSettings: dict
  Tags:
    - Key: string
      Value: string
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
  Args:
```

```
- string
OnNodeConfigured:
  Sequence:
    - Script: string
      Args:
        - string
    Script: string
    Args:
      - string
Iam:
  InstanceProfile: string
  InstanceRole: string
  S3Access:
    - BucketName: string
      EnableWriteAccess: boolean
      KeyName: string
  AdditionalIamPolicies:
    - Policy: string
Image:
  CustomAmi: string
```

### Scheduling:

```
Scheduler: awsbatch
AwsBatchQueues:
  - Name: string
  CapacityType: string
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
ComputeResources: # this maps to a Batch compute environment (initially we
support only 1)
  - Name: string
    InstanceTypes:
      - string
    MinvCpus: integer
    DesiredvCpus: integer
    MaxvCpus: integer
    SpotBidPercentage: float
```

## Scheduling 屬性

### Scheduler ( 必填 , String )

指定所使用的排程器類型。支援的值為 `slurm` 和 `awsbatch`。

更新原則：如果變更此設定，則不允許更新。

#### Note

`awsbatch` 僅支持 `alinux2` 操作系統和 `x86_64` 平台。

### ScalingStrategy ( 可選 , String )

可讓您選擇動態 Slurm 節點的擴充方式。支援的值為 `all-or-nothing` , `greedy-all-or-nothing` 且 `best-effort` 預設值為 `all-or-nothing`。

更新原則：您可以在更新期間變更此設定。

#### Note

擴展策略僅適用於 Slurm 要恢復的節點，而不適用於最終已在執行的節點。

- `all-or-nothing` 此策略嚴格遵循 `all-or-nothing-approach`，旨在避免在擴展過程結束時出現閒置實例。它運行的 `all-or-nothing` 基礎上，這意味著它要么完全擴展或根本沒有擴展。請注意，當工作需要 500 個以上的節點或跨越多個運算資源時，暫時啟動的執行個體可能會產生額外費用。在三種可能的擴展策略中，此策略的輸送量最低。縮放時間取決於每次執行 Slurm 恢復程式所提交的工作數目。此外，您不能擴展到遠遠超過每次執行的預設 `RunInstances` 資源帳號限制，預設為 1000 個執行個體。如需更多詳細資訊，請參閱 [Amazon EC2 API 節流文件](#)
- `greedy-all-or-nothing` 與 `all-or-nothing` 策略類似，它旨在避免進行縮放後的閒置實例。此策略允許在擴展過程中暫時過度擴展，以實現比 `all-or-nothing` 方法更高的輸送量，但也具有與 `RunInstances` 資源帳號限制相同的 1000 個執行個體擴展限制。
- `best-effort` 此策略會優先考慮高輸送量，即使這表示某些執行個體在擴展程序結束時可能處於閒置狀態。它嘗試根據作業要求分配盡可能多的節點，但有可能不滿足整個請求。與其他策略不同的是，最佳方法可以累積比標準 `RunInstances` 限制更多的執行個體，代價是在多個擴展程序執行中擁有閒置資源。

每種策略都是為了滿足不同的擴展需求而設計，允許您選擇一種符合您特定需求和限制的策略。

## AwsBatchQueues

(選擇性) AWS Batch 佇列設定。僅支援一個佇列。如果 [Scheduler](#) 設定為 `awsbatch`，則需要此區段。如需有關 `awsbatch` 排程器的詳細資訊，請參閱 [網路設定](#) 和 [AWS Batch \(awsbatch\)](#)。

```
AwsBatchQueues:
- Name: string
  CapacityType: string
  Networking:
    SubnetIds:
      - string
    AssignPublicIp: boolean
    SecurityGroups:
      - string
    AdditionalSecurityGroups:
      - string
  ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
- Name: string
  InstanceTypes:
    - string
  MinvCpus: integer
  DesiredvCpus: integer
  MaxvCpus: integer
  SpotBidPercentage: float
```

更新原則：您可以在更新期間變更此設定。

### AwsBatchQueues 屬性

#### Name ( 必填 , String )

AWS Batch 佇列的名稱。

更新原則：如果變更此設定，則不允許更新。

#### CapacityType ( 可選 , String )

AWS Batch 佇列使用的計算資源類型。支援的值為 `ONDEMAND`、`SPOT` 或 `CAPACITY_BLOCK`。預設值為 `ONDEMAND`。

**Note**

如果設定CapacityType為SPOT，您的帳戶必須包含AWSServiceRoleForEC2Spot服務連結角色。您可以使用以下 AWS CLI 命令創建此角色。

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

如需詳細資訊，請參閱 [Amazon Amazon EC2 Linux 執行個體使用者指南中的競價型執行個體請求的服務連結角色](#)。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

## Networking

(必要) 定義 AWS Batch 佇列的網路組態。

**Networking:****SubnetIds:**

- *string*

**AssignPublicIp:** *boolean***SecurityGroups:**

- *string*

**AdditionalSecurityGroups:**

- *string*

### Networking 屬性

#### SubnetIds ( 必填 , [String] )

指定要佈建 AWS Batch 佇列的現有子網路識別碼。目前只支援一個子網路。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

#### AssignPublicIp ( 可選 , String )

建立或指派公用 IP 位址給 AWS Batch 佇列中的節點。支援的值為 true 和 false。預設值取決於您指定的子網路。

更新原則：如果變更此設定，則不允許更新。

## SecurityGroups ( 可選 , [String] )

AWS Batch 佇列使用的安全性群組清單。如果您未指定安全性群組，請 AWS ParallelCluster 建立新的安全性群組。

更新原則：您可以在更新期間變更此設定。

## AdditionalSecurityGroups ( 可選 , [String] )

AWS Batch 佇列使用的安全性群組清單。

更新原則：您可以在更新期間變更此設定。

## ComputeResources

(必要) 定義 AWS Batch 佇列的 ComputeResources 組態。

```
ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
- Name: string
  InstanceTypes:
    - string
  MinvCpus: integer
  DesiredvCpus: integer
  MaxvCpus: integer
  SpotBidPercentage: float
```

## ComputeResources 屬性

### Name ( 必填 , String )

AWS Batch 佇列計算環境的名稱。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### InstanceTypes ( 必填 , [String] )

執行個體類型的 AWS Batch 運算環境陣列。所有執行個體類型都必須使用x86\_64架構。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### MinvCpus ( 可選 , Integer )

AWS Batch 計算環境可以使用的 vCPU 數目下限。

更新原則：您可以在更新期間變更此設定。

## DesiredVcpus ( 可選 , Integer )

AWS Batch 運算環境中所需的 vCPU 數目。AWS Batch MaxvCpus 根據工作佇列中的需求，在 MinvCpus 和之間調整此值。

更新原則：更新期間不會分析此設定。

## MaxvCpus ( 可選 , Integer )

AWS Batch 運算環境中的 vCPU 數目上限。您無法將此值設定為低於 DesiredVcpus。

更新原則：更新期間無法降低此設定。

## SpotBidPercentage ( 可選 , Float )

Amazon EC2 Spot 執行個體價格在執行個體啟動之前可達到的執行個體類型隨需價格的最高百分比。預設值為 100 (100%)。支援的範圍為 1-100。

更新原則：您可以在更新期間變更此設定。

## SlurmQueues

(選擇性) Slurm 佇列的設定。如果 [Scheduler](#) 設定為 slurm，則需要此區段。

```
SlurmQueues:
- Name: string
  ComputeSettings:
    LocalStorage:
      RootVolume:
        Size: integer
        Encrypted: boolean
        VolumeType: string
        Iops: integer
        Throughput: integer
      EphemeralVolume:
        MountDir: string
    CapacityReservationTarget:
      CapacityReservationId: string
      CapacityReservationResourceGroupArn: string
    CapacityType: string
    AllocationStrategy: string
    JobExclusiveAllocation: boolean
    CustomSlurmSettings: dict
  Tags:
    - Key: string
```



```
    Value: string
HealthChecks:
  Gpu:
    Enabled: boolean
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
  PlacementGroup:
    Enabled: boolean
    Id: string
    Name: string
  Proxy:
    HttpProxyAddress: string
ComputeResources:
  - Name: string
    InstanceType: string
    Instances:
      - InstanceType: string
    MinCount: integer
    MaxCount: integer
    DynamicNodePriority: integer
    StaticNodePriority: integer
    SpotPrice: float
    DisableSimultaneousMultithreading: boolean
    SchedulableMemory: integer
    HealthChecks:
      Gpu:
        Enabled: boolean
      Efa:
        Enabled: boolean
        GdrSupport: boolean
    CapacityReservationTarget:
      CapacityReservationId: string
      CapacityReservationResourceGroupArn: string
    Networking:
      PlacementGroup:
        Enabled: boolean
        Name: string
    CustomSlurmSettings: dict
```

```
Tags:
  - Key: string
    Value: string
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
        Script: string
        Args:
          - string
    OnNodeConfigured:
      Sequence:
        - Script: string
          Args:
            - string
          Script: string
          Args:
            - string
Iam:
  InstanceProfile: string
  InstanceRole: string
  S3Access:
    - BucketName: string
      EnableWriteAccess: boolean
      KeyName: string
  AdditionalIamPolicies:
    - Policy: string
Image:
  CustomAmi: string
```

更新原則：對於此清單值設定，可以在更新期間新增新值，或者在移除現有值時必須停止運算叢集。

## SlurmQueues 屬性

### Name ( 必填 , String )

Slurm佇列的名稱。

#### Note

叢集大小可能會在更新期間變更。如需詳細資訊，請參閱[叢集容量大小和更新](#)

更新原則：如果變更此設定，則不允許更新。

## CapacityReservationTarget

### Note

CapacityReservationTarget與 AWS ParallelCluster 版本 3.3.0 一起添加。

### CapacityReservationTarget:

[CapacityReservationId](#): *string*

[CapacityReservationResourceGroupArn](#): *string*

指定佇列計算資源的隨需容量保留。

### CapacityReservationId ( 可選 , String )

要針對佇列計算資源作為目標的現有容量保留區識別碼。識別碼可以參考 [OCR](#) 或 [ML 的容量區塊](#)。

保留項目必須使用執行個體使用的相同平台。例如，如果您的執行個體在上執行 `rhel8`，您的容量保留必須在 RHEL 平台上執行。如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的 [支援平台](#)。

### Note

如果您包含 [Instances](#) 在叢集配置中，則必須從組態中排除此佇列層級 CapacityReservationId 設定。

### CapacityReservationResourceGroupArn ( 可選 , String )

資源群組的 Amazon 資源名稱 (ARN)，用作佇列運算資源的服務連結容量保留群組。AWS ParallelCluster 根據下列條件，識別並使用資源群組中最適當的產能保留：

- 如果 PlacementGroup 在 [SlurmQueues/Networking](#) 或 [SlurmQueuesComputeResources](#) 中啟用 [Networking](#)，則 AWS ParallelCluster 選取以執行個體類型 PlacementGroup 為目標的資源群組，如果計算資源存在，則選取計算資源的資源群組。

PlacementGroup 必須以中定義的其中一個執行個體類型為目標 [ComputeResources](#)。

- 如果PlacementGroup未在 [SlurmQueues/Networking](#) 或 [SlurmQueuesComputeResources](#) 中啟用 [Networking](#)，則 AWS ParallelCluster 選取僅針對計算資源執行個體類型的資源群組 (如果計算資源存在)。

資源群組必須在佇列的所有計算資源和可用區域中保留在可用區域中的每個執行個體類型至少一個 ODCR。如需詳細資訊，請參閱 [使用隨需容量保留 \(ODCR\) 啟動執行個體](#)。

如需有關多個子網路組態需求的詳細資訊，請參閱 [Networking/SubnetIds](#)。

#### Note

在 AWS ParallelCluster 版本 3.4.0 中添加了多個可用區域。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### CapacityType (可選, String)

Slurm佇列使用的計算資源類型。支援的值為ONDEMAND、SPOT或CAPACITY\_BLOCK。預設值為ONDEMAND。

#### Note

如果將設定CapacityType為SPOT，您的帳戶必須具有AWSServiceRoleForEC2Spot服務連結角色。您可以使用以下 AWS CLI 命令創建此角色。

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

如需詳細資訊，請參閱 [Amazon Amazon EC2 Linux 執行個體使用者指南中的競價型執行個體請求的服務連結角色](#)。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### AllocationStrategy (可選, String)

為中定義的所有計算資源指定配置策略[Instances](#)。

有效值：lowest-price | capacity-optimized | price-capacity-optimized

預設：lowest-price

### lowest-price

- 如果您使用CapacityType = ONDEMAND，Amazon EC2 叢集會使用價格決定訂單，並先啟動最低價格的執行個體。
- 如果您使用CapacityType = SPOT，Amazon EC2 叢集會從具有可用容量的最低價格競價型執行個體集區啟動執行個體。如果集區在滿足所需容量之前耗盡容量，Amazon EC2 Fleet 會為您啟動執行個體來滿足您的請求。特別是，Amazon EC2 叢集會從具有可用容量的最低價格競價型執行個體集區啟動執行個體。Amazon EC2 叢集可能會從數個不同的集區啟動競價型執行個體。
- 如果您設定CapacityType = CAPACITY\_BLOCK，則沒有配置策略，因此無法配置AllocationStrategy參數。

### capacity-optimized

- 如果您已設定CapacityType = ONDEMAND，則capacity-optimized無法使用。
- 如果您已設定CapacityType = SPOT，Amazon EC2 叢集會從競價型執行個體集區啟動執行個體，並具有最佳容量以供啟動的執行個體數量使用。

### price-capacity-optimized

- 如果您已設定CapacityType = ONDEMAND，則capacity-optimized無法使用。
- 如果您設定CapacityType = SPOT，Amazon EC2 叢集會針對正在啟動的執行個體數量識別具有最高容量可用性的集區。這表示我們將從我們認為在短期內中斷機會最低的集區請求 Spot 執行個體。然後，Amazon EC2 叢集會從這些集區中最低價格請求競價型執行個體。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

#### Note

AllocationStrategy從 AWS ParallelCluster 版本 3.3.0 開始支援。

### JobExclusiveAllocation ( 可選 , String )

如果設定為true，則Slurm分割區OverSubscribe旗標會設定為EXCLUSIVE。當OverSubscribe = 時EXCLUSIVE，分割區中的作業對所有配置的節點具有獨佔存取權。[如需詳細資訊，請參閱Slurm文件中的 EXCLUSIVE。](#)

有效值：true | false

預設：false

更新原則：您可以在更新期間變更此設定。

**Note**

JobExclusiveAllocation從 AWS ParallelCluster 版本 3.7.0 開始受到支援。

## CustomSlurmSettings ( 可選 , Dict )

定義自訂Slurm分割區 (佇列) 組態設定。

指定套用至佇列 (分割區) 之自訂Slurm組態參數索引鍵值配對的字典。

每個單獨的索引鍵值對 (例如Param1: Value1) 會以格式Param1=Value1分別新增至Slurm分割區組態行的末端。

您只能指Slurm定未在中拒絕列出的組態參數。CustomSlurmSettings如需有關拒絕列出的Slurm組態參數的資訊，請參閱 [拒絕列出Slurm組態參數CustomSlurmSettings](#)

AWS ParallelCluster 僅檢查參數是否在拒絕清單中。AWS ParallelCluster 不會驗證您的自訂Slurm組態參數語法或語意。您必須負責驗證您的自訂Slurm組態參數。無效的自訂Slurm組態參數可能會造成Slurm協助程式失敗，進而導致叢集建立和更新失敗。

如需如何使用指定自訂Slurm組態參數的詳細資訊 AWS ParallelCluster，請參閱[Slurm組態自訂](#)。

如需有關Slurm組態參數的詳細資訊，請參閱說明文件中的 [slurm.conf](#)。Slurm

更新原則：您可以在更新期間變更此設定。

**Note**

CustomSlurmSettings從 AWS ParallelCluster 版本 3.6.0 開始支援。

## Tags ( 可選 , [字符串] )

標籤鍵值配對的清單。 [ComputeResource](#) 標籤會覆寫在 [Tags 區段](#) 或 SlurmQueues/ 中指定的重複標籤Tags。

## Key ( 可選 , String )

標籤金鑰。

## Value ( 可選 , String )

標籤值。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## HealthChecks(選擇性)

針對佇列中的所有計算資源指定計算節點健康狀態檢查。

## Gpu(選擇性)

針對佇列中的所有運算資源指定 GPU 健康狀態檢查。

### Note

AWS ParallelCluster 在使用 alinux2 ARM 作業系統Gpu的節點中不支援HealthChecks/。這些平台不支援 [NVIDIA 資料中心 GPU 管理員 \(DCGM\)](#)。

## Enabled ( 可選 , Boolean )

是否在運算節點上 AWS ParallelCluster 執行 GPU 健康狀態檢查。預設值為 false。

## Gpu健康檢查行為

- 如果設定Enabled為Gpu/true，則會對佇列中的計算資源 AWS ParallelCluster 執行 GPU 健康狀態檢查。
- Gpu健康狀態檢查會對運算資源執行 GPU 健康狀態檢查，以防止在具有降級 GPU 的節點上提交工作。
- 如果計算節點未通過Gpu健康狀態檢查，則計算節點狀態會變更為DRAIN。新工作不會在此節點上啟動。現有工作會執行至完成。所有執行中的作業完成後，如果計算節點是動態節點，則會終止運算節點，如果它是靜態節點，則會被取代。
- Gpu健全狀況檢查的持續時間取決於選取的執行個體類型、執行個體中的 GPU 數目，以及Gpu健全狀況檢查目標的數目 (相當於工作 GPU 目標的數目)。對於具有 8 個 GPU 的執行個體，一般持續時間不到 3 分鐘。
- 如果Gpu健康狀態檢查在不受支援的執行個體上執行，則會結束並在計算節點上執行工作。例如，如果執行個體沒有 GPU，或者執行個體具有 GPU，但它不是 NVIDIA GPU，則運作狀態檢查就會結束，而工作會在運算節點上執行。僅支援 NVIDIA 顯示卡。
- Gpu健全狀況檢查會使用此dcgmi工具對節點執行健全狀況檢查，並執行下列步驟：

在節點中開始Gpu健康狀態檢查時：

1. 它會偵測nvidia-dcgm和nvidia-fabricmanager服務是否正在執行。
2. 如果這些服務未執行，則Gpu健康狀態檢查會啟動它們。
3. 它檢測持久性模式是否已啟用。
4. 如果未啟用持續性模式，Gpu健康狀態檢查會啟用它。

健全狀況檢查結束時，健全狀Gpu況檢查會將這些服務和資源還原為其初始狀態。

- 如果將工作指派給一組特定的節點 GPU，則Gpu健全狀況檢查只會在該特定集上執行。否則，Gpu健全狀況檢查會在節點中的所有 GPU 上執行。
- 如果計算節點同時收到 2 個或多個Gpu健康狀態檢查要求，則只會執行第一個健全狀況檢查，而其他節點則會略過。以節點 GPU 為目標的運作狀態檢查也是這種情況。您可以檢查記錄檔，以取得有關此情況的其他資訊。
- 檔案中提供特定計算節點的健全狀況檢查記/var/log/parallelcluster/slurm\_health\_check.log錄。該文件在 Amazon 中可用 CloudWatch，在集群 CloudWatch 日誌組中，您可以在其中找到：
  - Gpu健全狀況檢查所執行動作的詳細資訊，包括啟用和停用服務和持續性模式。
  - GPU 識別碼、序列識別碼和 UUID。
  - 健康狀態檢查輸出。

更新原則：您可以在更新期間變更此設定。

#### Note

HealthChecks從 AWS ParallelCluster 版本 3.6.0 開始支援。

## Networking

(必要) 定義Slurm佇列的網路組態。

Networking:

SubnetIds:

- *string*

AssignPublicIp: *boolean*

SecurityGroups:

- *string*



**AdditionalSecurityGroups:**

- *string*

**PlacementGroup:**

**Enabled:** *boolean*

**Id:** *string*

**Name:** *string*

**Proxy:**

**HttpProxyAddress:** *string*

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## Networking 屬性

### SubnetIds ( 必填, [String] )

佈建Slurm佇列之現有子網路的 ID。

如果您在 [SlurmQueues/ComputeResources](#) 中設定執行個體類型 [InstanceType](#)，則只能定義一個子網路。

如果您在 [SlurmQueues/ComputeResources](#) 中設定執行個體類型 [Instances](#)，您可以定義單一子網路或多個子網路。

如果您使用多個子網路，則為佇列定義的所有子網路必須位於同一個 VPC 中，且每個子網路都位於不同的可用區域 (AZ) 中。

例如，假設您為佇列定義子網路 1 和子網路 2。

subnet-1並且不subnet-2能兩者都在 AZ-1 中。

subnet-1可以在 AZ-1 中，並且subnet-2可以在 AZ-2 中。

如果您只設定一個執行個體類型，而且想要使用多個子網路，請在中定義您的執行個體類型，Instances而InstanceType不是。

例如，定義ComputeResources/Instances/InstanceType= instance.type 而不是ComputeResources/InstanceType= instance.type。

#### Note

不同的可用區域不支援 Elastic Fabric Adapter (EFA)。

使用多個可用區域可能會導致儲存網路延遲增加，並增加 AZ 間資料傳輸成本。例如，當執行個體存取位於不同 AZ 中的檔案儲存時，就可能會發生這種情況。有關詳情，請參閱[相同內容的資料傳輸 AWS 區域](#)。

叢集更新以從使用單一子網路變更為多個子網路：

- 假設叢集的子網路定義是使用單一子網路和 Lustre 檔案系統的 AWS ParallelCluster 受管理 FSx 來定義。然後，您無法直接使用更新的子網路 ID 定義來更新此叢集。若要進行叢集更新，您必須先將受管理的檔案系統變更為外部檔案系統。如需詳細資訊，請參閱[將 AWS ParallelCluster 受管儲存轉換為外部儲存](#)。
- 假設叢集的子網路定義是以單一子網路和外部 Amazon EFS 檔案系統定義的，如果要新增的多個子網路的所有 AZ 都不存在 EFS 掛載目標。然後，您無法直接使用更新的子網路 ID 定義來更新此叢集。若要更新叢集或建立叢集，您必須先為已定義的多個子網路的所有 AZ 建立所有裝載目標。

可用區域和叢集容量保留定義於 [CapacityReservationResourceGroupArn](#)：

- 如果已定義的容量保留資源群組所涵蓋的執行個體類型集與可用區域集與為佇列定義的執行個體類型和可用區域集之間沒有重疊，則無法建立叢集。
- 如果已定義的容量保留資源群組所涵蓋的執行個體類型集與可用區域之間存在部分重疊，以及為佇列定義的執行個體類型和可用區域集，您可以建立叢集。AWS ParallelCluster 針對此情況傳送有關部分重疊的警告訊息。
- 如需詳細資訊，請參閱[使用隨需容量保留 \(ODCR\) 啟動執行個體](#)。

#### Note

在 AWS ParallelCluster 版本 3.4.0 中添加了多個可用區域。

#### Warning

此警告適用於 3.3.1 版之前的所有 3.x.y AWS ParallelCluster 版本。AWS ParallelCluster 如果變更此參數，版本 3.3.1 不會受到影響。

對於 3.3.1 版之前的 AWS ParallelCluster 3 個版本：

變更此參數並更新叢集會為 Lustre 檔案系統建立新的受管理 FSx，並刪除 Lustre 檔案系統的現有受管理 FSx，而不保留現有資料。這會導致數據丟失。在繼續之前，如果您要保留資料，請確定已備份來自現有 FSx for Lustre 檔案系統的資料。如需詳細資訊，請參閱《FSx for Lustre 使用者指南》中的[〈使用備份〉](#)。

如果新增子網路值，[更新原則：您可以在更新期間變更此設定。](#)

如果子網路值已移除，[更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。](#)

### AssignPublicIp (可選, String)

建立或指派公用 IP 位址給Slurm佇列中的節點。支援的值為 true 和 false。您指定的子網路會決定預設值。具有公用 IP 的子網路預設為指派公用 IP 位址。

如果您定義p4d或hpc6id執行個體類型，或其他具有多個網路介面或網路介面卡的執行個體類型，則必須將 [HeadNode/Networking/設定ElasticIp](#) true 為提供公用存取權。AWS 只能將公有 IP 指派給透過單一網路介面啟動的執行個體。在此情況下，建議您使用 [NAT 閘道](#) 來提供叢集運算節點的公用存取權。在此情況下，AssignPublicIp請將設定為false。[如需 IP 地址的詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公有 IPv4 地址。](#)

[更新原則：如果變更此設定，則不允許更新。](#)

### SecurityGroups (可選, [String])

要用於Slurm佇列的安全性群組清單。如果未指定安全性群組，請為您 AWS ParallelCluster 建立安全性群組。

確認您的[SharedStorage](#)系統已正確設定安全群組。

#### Warning

此警告適用於所有 3. *x.y* AWS ParallelCluster 版本 3.3.0 之前的版本。AWS ParallelCluster 如果更改此參數，3.3.0 版本不會受到影響。

對於版本 3.3.0 之前的 AWS ParallelCluster 3 個版本：

變更此參數並更新叢集會為 Lustre 檔案系統建立新的受管理 FSx，並刪除 Lustre 檔案系統的現有受管理 FSx，而不保留現有資料。這會導致數據丟失。如果您要保留資料，請務必備份 Lustre 檔案系統的現有 FSx 中的資料。如需詳細資訊，請參閱《FSx for Lustre 使用者指南》中的 [〈使用備份〉](#)。

#### Warning

如果您為運算執行個體啟用 [Efa](#)，請確定已啟用 EFA 的執行個體屬於安全群組的成員，該群組允許所有輸入和輸出流量傳入本身。

更新原則：您可以在更新期間變更此設定。

### **AdditionalSecurityGroups** ( 可選 , [String] )

要用於Slurm佇列的其他安全性群組清單。

更新原則：您可以在更新期間變更此設定。

### **PlacementGroup**(選擇性)

指定Slurm佇列的放置群組設定。

```
PlacementGroup:  
  Enabled: boolean  
  Id: string  
  Name: string
```

更新原則：必須停止所有運算節點，才能刪除受管理的置放群組。必須停止或設定計算叢集，QueueUpdateStrategy才能變更此設定以進行更新。

### **Enabled** ( 可選 , Boolean )

指出Slurm佇列是否使用放置群組。預設值為 false。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### **Id** ( 可選 , String )

Slurm佇列使用之現有叢集置放群組的放置群組名稱。請務必提供放置群組名稱，而非 ID。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### **Name** ( 可選 , String )

Slurm佇列使用之現有叢集置放群組的放置群組名稱。請務必提供放置群組名稱，而非 ID。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

#### Note

- 如果將 PlacementGrouptrue/設定Enabled為 (未Id定義Name或)，則會為每個計算資源指派自己的受管理放置群組，除非PlacementGroup已定義ComputeResourcesNetworking//來覆寫此設定。

- 從 AWS ParallelCluster 版本 3.3.0 開始，[SlurmQueues/NetworkingPlacementGroup/Name](#) 被添加為 [SlurmQueues/Networking](#) 的首選替代方案。 [PlacementGroupId](#) [PlacementGroup/Id](#) 和 [PlacementGroup/Name](#) 是等價的。您可以使用任何一種。如果您同時包含 [PlacementGroup/Id](#) 和 [PlacementGroup/Name](#)，則 AWS ParallelCluster 失敗。您只能選擇一個或另一個。您不需要更新叢集即可使用 [PlacementGroup/Name](#)。

## Proxy(選擇性)

指定Slurm佇列的代理伺服器設定。

```
Proxy:  
  HttpProxyAddress: string
```

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### HttpProxyAddress ( 可選 , String )

定義Slurm佇列的 HTTP 或 HTTPS 代理伺服器。通常情況下，它是 `https://x.x.x.x:8080`。

沒有預設值。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## Image

(選擇性) 指定要用於Slurm佇列的影像。若要對所有節點使用相同的 AMI，請使用 [Image](#) 區段中的 [CustomAmi](#) 設定。

```
Image:  
  CustomAmi: string
```

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## Image性質

### CustomAmi ( 可選 , String )

用於Slurm佇列而非預設 AMI 的 AMI。您可以使用 pcluster CLI 命令來檢視預設 AMI 的清單。

#### Note

AMI 必須以頭節點所使用的相同作業系統為基礎。

```
pcluster list-official-images
```

如果自訂 AMI 需要其他權限才能啟動，您必須將這些權限新增至頭節點原則。

例如，如果自訂 AMI 具有與其關聯的加密快照，則標頭節點原則中需要下列其他原則。

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

若要疑難排解自訂 AMI 驗證警告，請參閱[排解自訂 AMI 問題](#)。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## ComputeResources

(必要) 定義Slurm佇列的ComputeResources組態。

### Note

叢集大小可能會在更新期間變更。如需詳細資訊，請參閱[叢集容量大小和更新](#)

#### ComputeResources:

- Name: *string*
- InstanceType: *string*
- Instances:
  - InstanceType: *string*
- MinCount: *integer*
- MaxCount: *integer*
- DynamicNodePriority: *integer*
- StaticNodePriority: *integer*
- SpotPrice: *float*
- DisableSimultaneousMultithreading: *boolean*
- SchedulableMemory: *integer*
- HealthChecks:
  - Gpu:
    - Enabled: *boolean*
- Efa:
  - Enabled: *boolean*
  - GdrSupport: *boolean*
- CapacityReservationTarget:
  - CapacityReservationId: *string*
  - CapacityReservationResourceGroupArn: *string*
- Networking:
  - PlacementGroup:
    - Enabled: *boolean*
    - Name: *string*
- CustomSlurmSettings: *dict*
- Tags:
  - Key: *string*
  - Value: *string*

更新原則：對於此清單值設定，可以在更新期間新增新值，或者在移除現有值時必須停止運算叢集。

## ComputeResources 屬性

### Name ( 必填 , String )

Slurm佇列計算環境的名稱。名稱最多可包含 25 個字元。

更新原則：如果變更此設定，則不允許更新。

### InstanceType ( 必填 , String )

此Slurm計算資源中使用的執行個體類型。叢集中的所有執行個體類型都必須使用相同的處理器架構。執行個體可以使用x86\_64或arm64架構。

叢集配置必須定義[InstanceType](#)或[執行個體](#)。如果兩者都已定義，則 AWS ParallelCluster 失敗。

定義時InstanceType，您無法定義多個子網路。如果您只設定一個執行個體類型，而且想要使用多個子網路，請在中定義您的執行個體類型，Instances而不是在中InstanceType定義。如需詳細資訊，請參閱 [Networking/SubnetIds](#)。

如果您定義p4d或hpc6id執行個體類型，或其他具有多個網路介面或網路介面卡的執行個體類型，則必須在私有子網路中啟動運算執行個體，如中所述[AWS ParallelCluster 使用兩個子網](#)。AWS 只能將公有 IP 指派給透過單一網路介面啟動的執行個體。[如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公用 IPv4 地址](#)。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### Instances(必填)

指定計算資源的執行個體類型清單。若要指定執行個體類型清單的配置策略，請參閱[AllocationStrategy](#)。

叢集配置必須定義[InstanceType](#)或[Instances](#)。如果兩者都已定義，則 AWS ParallelCluster 失敗。

如需詳細資訊，請參閱 [使用 Slurm 進行多個實例類型分配](#)。

#### [Instances](#):

- [InstanceType](#): *string*

#### Note

從 AWS ParallelCluster 版本 3.7.0 開始，如果您在執行個體中設定多個[執行個體類型](#)，則EnableMemoryBasedScheduling可以啟用此功能。



適用於 AWS ParallelCluster 版本 3.2.0 至 3.6 版本。

**x**，`EnableMemoryBasedScheduling` 如果您在執行個體中設定多個執行個體類型，則無法啟用。

更新原則：對於此清單值設定，可以在更新期間新增新值，或者在移除現有值時必須停止運算叢集。

### InstanceType ( 必填, String )

要在此 Slurm 計算資源中使用的執行個體類型。叢集中的所有執行個體類型都必須使用相同的處理器架構 (x86\_64 或) arm64。

中列出的執行個體類型 [Instances](#) 必須具有：

- 相同數量的 vCPUs，或者，如果設定 `DisableSimultaneousMultithreading` 為 true，則相同數量的核心。
- 相同數量的同一製造商的加速器。
- EFA 支援 (如果 `Efa/Enabled` 設定為 true)。

中列出的執行個體類型 [Instances](#) 可以具有：

- 不同的記憶體容量。

在這種情況下，最小內存將被設置為可消耗 Slurm 資源。

#### Note

從 AWS ParallelCluster 版本 3.7.0 開始，如果您在執行個體中設定多個執行個體類型，則 `EnableMemoryBasedScheduling` 可以啟用此功能。

適用於 AWS ParallelCluster 版本 3.2.0 至 3.6 版本。

**x**，`EnableMemoryBasedScheduling` 如果您在執行個體中設定多個執行個體類型，則無法啟用。

- 不同的網卡。

在這種情況下，為計算資源設定的網路介面數目是由具有最少網路卡數目的執行個體類型所定義。

- 不同的網絡帶寬。
- 不同的實例存儲大小。

如果您定義p4d或hpc6id執行個體類型，或其他具有多個網路介面或網路介面卡的執行個體類型，則必須在私有子網路中啟動運算執行個體，如中所述[AWS ParallelCluster 使用兩個子網](#)。AWS 只能將公有 IP 指派給透過單一網路介面啟動的執行個體。[如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動期間指派公用 IPv4 地址](#)。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

**Note**

Instances從 AWS ParallelCluster 版本 3.3.0 開始支援。

### MinCount ( 可選 , Integer )

Slurm運算資源使用的執行個體數目下限。預設值為 0。

**Note**

叢集大小可能會在更新期間變更。如需詳細資訊，請參閱[叢集容量大小和更新](#)

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### MaxCount ( 可選 , Integer )

Slurm運算資源使用的執行個體數目上限。預設為 10。

使用時CapacityType = CAPACITY\_BLOCK，MaxCount必須等於或大於 0，因為容量區塊保留區的所有執行個體部分都會作為靜態節點進行管理。MinCount

在叢集建立時，頭節點會等待所有靜態節點準備就緒，然後才能成功建立叢集。但是，使用時CapacityType = CAPACITY\_BLOCK，此檢查不會考慮與容量區塊相關聯之計算資源的節點部分。即使並非所有已設定的容量區塊都處於作用中狀態，仍會建立叢集。

**Note**

叢集大小可能會在更新期間變更。如需詳細資訊，請參閱[叢集容量大小和更新](#)

## DynamicNodePriority ( 可選 , Integer )

佇列計算資源中動態節點的優先順序。優先順序會Slurm對應至計算資源動態節點的節點[Weight](#)組態參數。預設值為 1000。

Slurm優先排列具有最低Weight值的節點。

### Warning

在Slurm分割區 (佇列) 中使用許多不同的Weight值可能會降低佇列中工作排程的速率。在 AWS ParallelCluster 版本 3.7.0 之前的版本中，靜態和動態節點都被指派相同的1預設權重。在這種情況下，由於靜態和動態節點的命名結構描述，Slurm可能會將閒置動態節點優先於閒置的靜態節點。當所有其他方式相等時，會依名稱的字母順序排Slurm程節點。

### Note

DynamicNodePriority已在 AWS ParallelCluster 版本 3.7.0 中新增。

[更新原則：您可以在更新期間變更此設定。](#)

## StaticNodePriority ( 可選 , Integer )

佇列計算資源中靜態節點的優先順序。優先順序會Slurm對應至計算資源靜態節點的節點[Weight](#)組態參數。預設值為 1。

Slurm優先排列具有最低Weight值的節點。

### Warning

在Slurm分割區 (佇列) 中使用許多不同的Weight值可能會降低佇列中工作排程的速率。

### Note

StaticNodePriority已在 AWS ParallelCluster 版本 3.7.0 中新增。

[更新原則：您可以在更新期間變更此設定。](#)

## SpotPrice ( 可選 , Float )

任何執行個體啟動前，Amazon EC2 競價型執行個體支付的最高價格。預設值為隨需價格。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## DisableSimultaneousMultithreading ( 可選 , Boolean )

如果true，停用Slurm佇列中節點上的多執行緒。預設值為false。

並非所有執行個體類型都能停用多執行緒。如需支援停用多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](#)。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

## SchedulableMemory ( 可選 , Integer )

在計算資源的運算節點Slurm參數RealMemory中設定的 MiB 記憶體容量。這個值是啟用 [SlurmSettings/EnableMemoryBasedScheduling](#)時，工作可用的節點記憶體上限。預設值是 Amazon EC2 執行個體類型中列出且由 Amazon EC2 API 傳回的記憶體的 95% [DescribeInstanceTypes](#)。確保將 GiB 中給出的值轉換為 MiB。

支援的值：1-EC2Memory

EC2Memory是列在 Amazon EC2 執行個體類型中並由 Amazon EC2 API [DescribeInstanceTypes](#)傳回的記憶體 (以 MiB 為單位)。確保將 GiB 中給出的值轉換為 MiB。

啟用 [SlurmSettings/EnableMemoryBasedScheduling](#)時，此選項最相關。如需詳細資訊，請參閱 [Slurm記憶體式排程](#)。

### Note

SchedulableMemory從 AWS ParallelCluster 版本 3.2.0 開始支援。依預設，從版本 3.2.0 開始，會將Slurm運算節點設 AWS ParallelCluster 定RealMemory為 Amazon EC2 API 傳回的 95% 記憶體。DescribeInstanceTypes此組態與的值無關EnableMemoryBasedScheduling。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## HealthChecks(選擇性)

指定計算資源的健康狀態檢查。

### Gpu(選擇性)

針對計算資源指定 GPU 健康狀態檢查。

#### Enabled ( 可選 , Boolean )

是否 AWS ParallelCluster 對佇列中的資源執行 GPU 健康狀態檢查。預設值為 false。

#### Note

AWS ParallelCluster 在使用 alinux2 ARM 作業系統 Gpu 的節點中不支援 HealthChecks/。這些平台不支援 [NVIDIA 資料中心 GPU 管理員 \(DCGM\)](#)。

### Gpu健康檢查行為

- 如果將 Gpu/設定 Enabled 為 true，則會對運算資源 AWS ParallelCluster 執行健全狀況 GPU 健康狀態檢查。
- Gpu健康狀態檢查會對運算資源執行健康狀態檢查，以防止在具有降級 GPU 的節點上提交工作。
- 如果計算節點未通過 Gpu健康狀態檢查，則計算節點狀態會變更為 DRAIN。新工作不會在此節點上啟動。現有工作會執行至完成。所有執行中的作業完成後，如果計算節點是動態節點，則會終止運算節點，如果它是靜態節點，則會被取代。
- Gpu健全狀況檢查的持續時間取決於選取的執行個體類型、執行個體中的 GPU 數目，以及 Gpu健全狀況檢查目標的數目 (相當於工作 GPU 目標的數目)。對於具有 8 個 GPU 的執行個體，一般持續時間不到 3 分鐘。
- 如果 Gpu健康狀態檢查在不受支援的執行個體上執行，則會結束並在計算節點上執行工作。例如，如果執行個體沒有 GPU，或者執行個體具有 GPU，但它不是 NVIDIA GPU，則運作狀態檢查就會結束，而工作會在運算節點上執行。僅支援 NVIDIA 顯示卡。
- Gpu健全狀況檢查會使用此 dcmi 工具對節點執行健全狀況檢查，並執行下列步驟：

在節點中開始 Gpu健康狀態檢查時：

1. 它會偵測 nvidia-dcgm 和 nvidia-fabricmanager 服務是否正在執行。
2. 如果這些服務未執行，則 Gpu健康狀態檢查會啟動它們。

3. 它檢測持久性模式是否已啟用。
4. 如果未啟用持續性模式，Gpu健康狀態檢查會啟用它。

健全狀況檢查結束時，健全狀Gpu況檢查會將這些服務和資源還原為其初始狀態。

- 如果將工作指派給一組特定的節點 GPU，則Gpu健全狀況檢查只會在該特定集上執行。否則，Gpu健全狀況檢查會在節點中的所有 GPU 上執行。
- 如果計算節點同時收到 2 個或多個Gpu健康狀態檢查要求，則只會執行第一個健全狀況檢查，而其他節點則會略過。以節點 GPU 為目標的運作狀態檢查也是這種情況。您可以檢查記錄檔，以取得有關此情況的其他資訊。
- 檔案中提供特定計算節點的健全狀況檢查記/var/log/parallelcluster/slurm\_health\_check.log錄。您可以在 Amazon CloudWatch 的叢集 CloudWatch 日誌群組中找到此檔案：
  - Gpu健全狀況檢查所執行動作的詳細資訊，包括啟用和停用服務和持續性模式。
  - GPU 識別碼、序列識別碼和 UUID。
  - 健康狀態檢查輸出。

[更新原則：您可以在更新期間變更此設定。](#)

#### Note

HealthChecks從 AWS ParallelCluster 版本 3.6.0 開始支援。

## Efa(選擇性)

指定Slurm佇列中節點的 Elastic Fabric Adapter (EFA) 設定。

Efa:

Enabled: *boolean*

GdrSupport: *boolean*

[更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。](#)

### **Enabled** ( 可選 , **Boolean** )

指定啟用 Elastic Fabric Adapter (EFA)。若要檢視支援 EFA 的 Amazon EC2 執行個體清單，請參閱 Amazon EC2 執行個體使用者指南中支援的執行 [個體類型](#)。如需詳細資訊，請參閱

[Elastic Fabric Adapter](#)。建議您使用叢集 [SlurmQueues/Networking/](#)，將執行個體之間的延遲降 [PlacementGroup](#) 到最低。

預設值為 `false`。

#### Note

不同的可用區域不支援 Elastic Fabric Adapter (EFA)。如需詳細資訊，請參閱 [SubnetIds](#)。

#### Warning

如果您要在中定義自訂安全性群組 [SecurityGroups](#)，請確定已啟用 EFA 的執行個體是允許所有輸入和輸出流量傳輸到本身的安全性群組的成員。

更新原則：必須停止運算叢集，或者 `QueueUpdateStrategy` 必須設定，才能變更此設定以進行更新。

### **GdrSupport** ( 可選 , **Boolean** )

(選擇性) 從 AWS ParallelCluster 版本 3.0.2 開始，此設定沒有任何作用。如果 Slurm 運算資源和作業系統的執行個體類型支援，GPU Direct RDMA (遠端直接記憶體存取) 的彈性網狀架構介面卡 (EFA) 支援一律會啟用。

#### Note

AWS ParallelCluster 版本 3.0.0 到 3.0.1：已 Support 對運算資源啟用 GPU 直接 RDMA 的支援。Slurm 特定作業系統 (是 `alinux2`、`centos7`、或) 上的特定執行個體類型 (`p4d.24xlarge`) Support 援 GPU Direct RDMA [Os](#) 的支援。ubuntu1804 ubuntu2004 預設值為 `false`。

更新原則：必須停止運算叢集，或者 `QueueUpdateStrategy` 必須設定，才能變更此設定以進行更新。

### **CapacityReservationTarget**

```
CapacityReservationTarget:  
  CapacityReservationId: string
```

`CapacityReservationResourceGroupArn`: *string*

指定用於計算資源的隨需容量保留。

### `CapacityReservationId` (可選, `String`)

要針對佇列計算資源作為目標的現有容量保留區識別碼。識別碼可以參考 [OCR](#) 或 [ML 的容量區塊](#)。

在計算資源層級指定此參數時，`InstanceType`為選擇性參數，系統會自動從保留區擷取該參數。

### `CapacityReservationResourceGroupArn` (可選, `String`)

指示資源群組的 Amazon 資源名稱 (ARN)，該資源群組用作運算資源的服務連結容量保留群組。AWS ParallelCluster 識別並使用群組中最適當的容量保留。針對計算資源列出的每個執行個體類型，資源群組必須至少有一個 ODCR。如需詳細資訊，請參閱 [使用隨需容量保留 \(ODCR\) 啟動執行個體](#)。

- 如果 `PlacementGroup` 在 [SlurmQueues/Networking](#) 或 [SlurmQueues/ComputeResources](#) 中啟用 [Networking](#)，則 AWS ParallelCluster 選取以執行個體類型 `PlacementGroup` 為目標的資源群組，並選取計算資源 (如果存在)。

`PlacementGroup` 必須以中定義的其中一個例證類型為目標 [ComputeResources](#)。

- 如果 `PlacementGroup` 未在 [SlurmQueues/Networking](#) 或 [SlurmQueues/ComputeResources](#) 中啟用 [Networking](#)，則 AWS ParallelCluster 選取僅針對計算資源執行個體類型 (如果存在) 的資源群組。

更新原則：必須停止運算叢集，或者 `QueueUpdateStrategy` 必須設定，才能變更此設定以進行更新。

#### Note

`CapacityReservationTarget` 與 AWS ParallelCluster 版本 3.3.0 一起添加。

## Networking

[Networking](#):

[PlacementGroup](#):

[Enabled](#): *boolean*

[Name](#): *string*



更新原則：必須停止所有運算節點，才能刪除受管理的置放群組。必須停止或設定計算叢集，QueueUpdateStrategy才能變更此設定以進行更新。

### PlacementGroup(選擇性)

指定計算資源的置放群組設定。

#### Enabled ( 可選 , Boolean )

指出是否將置放群組用於計算資源。

- 如果設定為 true (未Name定義)，則無論使用 [SlurmQueues/Networking/PlacementGroup](#) 設定為何，都會為該計算資源指派自己的受管理置放群組。
- 如果設定為 (已定Name義)，則會將true該計算資源指派給具名的置放群組，而不考慮SlurmQueues/Networking/PlacementGroup設定。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

#### Name ( 可選 , String )

用於計算資源之現有叢集置放群組的置放群組名稱。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

#### Note

- 如果兩者都Name沒有設置PlacementGroup/Enabled，則其各自的值默認為 [SlurmQueues/Networking/PlacementGroup](#) 設置。
- ComputeResources/Networking/PlacementGroup添加了3.3.0 AWS ParallelCluster 版本。

### CustomSlurmSettings ( 可選 , Dict )

(選擇性) 定義自訂Slurm節點 (計算資源) 組態設定。

指定套用至Slurm節點 (計算資源) 的自訂Slurm組態參數鍵值配對的字典。

每個單獨的索引鍵值對 (例如Param1: Value1) 會以格式Param1=Value1分別新增至Slurm節點組態行的結尾。

您只能指Slurm定未在中拒絕列出的組態參數。CustomSlurmSettings如需有關拒絕列出的Slurm組態參數的資訊，請參閱。[拒絕列出Slurm組態參數CustomSlurmSettings](#)

AWS ParallelCluster 僅檢查參數是否在拒絕清單中。AWS ParallelCluster 不會驗證您的自訂Slurm組態參數語法或語意。您必須負責驗證您的自訂Slurm組態參數。無效的自訂Slurm組態參數可能會造成Slurm協助程式失敗，進而導致叢集建立和更新失敗。

如需如何使用指定自訂Slurm組態參數的詳細資訊 AWS ParallelCluster，請參閱[Slurm組態自訂](#)。

如需有關Slurm組態參數的詳細資訊，請參閱說明文件中的 [slurm.conf](#)。Slurm

[更新原則：您可以在更新期間變更此設定。](#)

#### Note

CustomSlurmSettings從 AWS ParallelCluster 版本 3.6.0 開始支援。

## Tags ( 可選 , [字符串] )

標籤鍵值配對的清單。ComputeResource標籤會覆寫[Tags 區段](#)或 [SlurmQueues](#)/中指定的重複標籤Tags。

### Key ( 可選 , String )

標籤金鑰。

### Value ( 可選 , String )

標籤值。

[更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。](#)

## ComputeSettings

(必要) 定義Slurm佇列的ComputeSettings組態。

### ComputeSettings 屬性

指定Slurm佇列ComputeSettings中節點的屬性。

[ComputeSettings](#):

```
LocalStorage:  
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  VolumeType: string  
  Iops: integer  
  Throughput: integer  
EphemeralVolume:  
  MountDir: string
```

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### LocalStorage(選擇性)

指定Slurm佇列LocalStorage中節點的屬性。

```
LocalStorage:  
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  VolumeType: string  
  Iops: integer  
  Throughput: integer  
EphemeralVolume:  
  MountDir: string
```

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### RootVolume(選擇性)

指定Slurm佇列中節點之根磁碟區的詳細資訊。

```
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  VolumeType: string  
  Iops: integer  
  Throughput: integer
```

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## Size ( 可選 , Integer )

指定佇列中節點的根磁碟區大小 (以 GiB) 為單位。Slurm預設大小來自 AMI。使用不同的大小需要 AMI 支援growroot。

[更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。](#)

## Encrypted ( 可選 , Boolean )

如果true，則會加密Slurm佇列中節點的根磁碟區。預設值為 false。

[更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。](#)

## VolumeType ( 可選 , String )

指定Slurm佇列中節點的 [Amazon EBS 磁碟區類型](#)。支援的值為gp2gp3、io1、io2、sc1、st1、和standard。預設值為 gp3。

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](#)。

[更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。](#)

## Iops ( 可選 , Boolean )

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VolumeType不同Size。

### VolumeType = io1

預設值 Iops =

支援的值 Iops =

最大volume\_iops比volume\_size率 = 每 GiB 50 IOPS。5000 IOPS 需要 -volume\_size個至少 100 GiB。

### VolumeType = io2

預設值 Iops =

支援的值 Iops = 100 至 64 萬 (對於io2區塊快速磁碟區為 256000) †

最大Iops比Size率 = 每 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB 博Size的一個。

### VolumeType = gp3

預設Iops值

支援的值 Iops =

對Iops於 IOPS 大於 3000 的磁碟區，最大對Size比例 = 每 GiB 500 IOPS。

† 只有在 [Nitro 系統上建置的執行個體 \(也佈建了超過 32,000 IOPS\) 上](#)，才能保證最大 IOPS。其他執行個體最多可以有 32,000 IOPS。除非您[修改io1磁碟區](#)，否則舊版磁碟區可能無法達到完整效能。io2區塊快速磁碟區在R5b執行個體類型上支援高達 256000 的volume\_iops值。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](#)。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### Throughput ( 可選 , Integer )

定義gp3磁碟區類型的輸送量，以 MIB/s 為單位。此設定只有在為時VolumeType才有效gp3。預設值為 125。支援的值：每秒

與的比率Iops可Throughput以不超過 0.25。每秒 1000 MB 的最大輸送量需要Iops設定至少為 4000。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### EphemeralVolume ( 可選 , Boolean )

指定暫時體積塊的設定。暫時磁碟區是透過將所有執行個體儲存磁碟區合併為使用檔案系統格式化的單一邏輯磁碟區來建立。ext4預設值為 /scratch。如果執行個體類型沒有任何執行個體儲存磁碟區，則不會建立暫時磁碟區。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的《[執行個體儲存體磁碟區](#)》。

EphemeralVolume:  
MountDir: *string*

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### MountDir ( 可選 , String )

佇列中每個節點之暫時磁碟區的掛載目錄。Slurm

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

## CustomActions

(選擇性) 指定要在Slurm佇列中的節點上執行的自訂指令碼。

```
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeConfigured:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
```

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### CustomActions性質

#### OnNodeStart ( 可選 , String )

指定啟動任何節點部署啟動程序動作之前，要在Slurm佇列中的節點上執行的一系列指令碼或單一指令碼。AWS ParallelCluster 不支持包括單個腳本和相同Sequence的自定義操作。如需詳細資訊，請參閱 [自訂引導動作](#)。

## Sequence(選擇性)

要執行的指令碼清單。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

**Script** ( 必填, **String** )

要使用的檔案。檔案路徑可以以https://或開頭s3://。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

**Args** ( 可選, [**String**] )

要傳遞給指令碼的引數清單。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

**Script** ( 必填, **String** )

要用於單一指令碼的檔案。檔案路徑可以以https://或開頭s3://。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

**Args** ( 可選, [**String**] )

要傳遞給單一指令碼的引數清單。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

**OnNodeConfigured** ( 可選, **String** )

指定所有節點啟動程序動作完成後，要在Slurm佇列中的節點上執行的指令碼序列或單一指令碼。AWS ParallelCluster 不支持包括單個腳本和相同Sequence的自定義操作。如需詳細資訊，請參閱[自訂引導動作](#)。

## Sequence(選擇性)

要執行的指令碼清單。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### Script ( 必填 , String )

要使用的檔案。檔案路徑可以以https://或開頭s3://。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### Args ( 可選 , [String] )

要傳遞給指令碼的引數清單。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### Script ( 必填 , String )

要用於單一指令碼的檔案。檔案路徑可以以https://或開頭s3://。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

### Args ( 可選 , [String] )

要傳遞給單一指令碼的引數清單。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

更新原則：必須停止運算叢集，或者QueueUpdateStrategy必須設定，才能變更此設定以進行更新。

#### Note

Sequence從 AWS ParallelCluster 版本 3.6.0 開始添加。當您指定時Sequence，您可以列出自訂動作的多個指令碼。AWS ParallelCluster 繼續支援使用單一指令碼設定自訂動作，但不包含Sequence。



AWS ParallelCluster 不支持包括單個腳本和相同Sequence的自定義操作。

## Iam

(選擇性) 為Slurm佇列定義選用的 IAM 設定。

```
Iam:
  S3Access:
    - BucketName: string
      EnableWriteAccess: boolean
      KeyName: string
  AdditionalIamPolicies:
    - Policy: string
  InstanceProfile: string
  InstanceRole: string
```

更新原則：您可以在更新期間變更此設定。

## Iam性質

### InstanceProfile ( 可選 , String )

指定執行個體設定檔，以覆寫Slurm佇列的預設執行個體角色或執行個體設定檔。您無法同時指定InstanceProfile和InstanceRole。格式是 `arn:${Partition}:iam::${Account}:instance-profile/${InstanceProfileName}`。

如果已指定，則無法指AdditionalIamPolicies定S3Access和設定。

建議您指定其中一個或兩個S3Access和AdditionalIamPolicies設定，因為新增至的功能AWS ParallelCluster 通常需要新的權限。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### InstanceRole ( 可選 , String )

指定執行個體角色，以覆寫Slurm佇列的預設執行個體角色或執行個體設定檔。您無法同時指定InstanceProfile和InstanceRole。格式是 `arn:${Partition}:iam::${Account}:role/${RoleName}`。

如果已指定，則無法指AdditionalIamPolicies定S3Access和設定。

建議您指定其中一個或兩個S3Access和AdditionalIamPolicies設定，因為新增至的功能AWS ParallelCluster 通常需要新的權限。

[更新原則：您可以在更新期間變更此設定。](#)

### S3Access(選擇性)

指定Slurm佇列的值區。這是用來產生政策，將指定存取權授與Slurm佇列中的值區。

如果已指定，則無法指定InstanceRole和InstanceProfile和設定。

建議您指定其中一個或兩個S3Access和AdditionalIamPolicies設定，因為新增至的功能AWS ParallelCluster 通常需要新的權限。

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

[更新原則：您可以在更新期間變更此設定。](#)

### BucketName ( 必填 , String )

儲存貯體的名稱。

[更新原則：您可以在更新期間變更此設定。](#)

### KeyName ( 可選 , String )

存儲桶的關鍵。預設值為 \*。

[更新原則：您可以在更新期間變更此設定。](#)

### EnableWriteAccess ( 可選 , Boolean )

指出是否已啟用值區的寫入存取權。

[更新原則：您可以在更新期間變更此設定。](#)

### AdditionalIamPolicies(選擇性)

為 Amazon Amazon EC2 指定 IAM 政策的亞馬遜資源名稱 (ARN) 清單。除了需要的權限之外，此清單還會附加至用於Slurm佇列的根角色 AWS ParallelCluster。

IAM 政策名稱及其 ARN 不同。不能使用名稱。

如果已指定，則無法指定 InstanceRole 和 InstanceProfile 設定。

我們建議您使用 `AdditionalIamPolicies` 為已新增至 AWS ParallelCluster 需要的權限，且 InstanceRole 必須包含所有必要的權限。所需的許可經常隨發行版本新增功能而變更。

沒有預設值。

AdditionalIamPolicies:

- Policy: *string*

更新原則：您可以在更新期間變更此設定。

**Policy** ( 必填, [String] )

IAM 政策清單。

更新原則：您可以在更新期間變更此設定。

## SlurmSettings

(選擇性) 定義套用至整個叢集的設定。Slurm

SlurmSettings:

ScaledownIdletime: *integer*

QueueUpdateStrategy: *string*

EnableMemoryBasedScheduling: *boolean*

CustomSlurmSettings: [*dict*]

CustomSlurmSettingsIncludeFile: *string*

Database:

Uri: *string*

UserName: *string*

PasswordSecretArn: *string*

ExternalSlurmdbd:

Host: *string*

Port: *integer*

Dns:

DisableManagedDns: *boolean*

HostedZoneId: *string*

`UseEc2Hostnames`: *boolean*

## SlurmSettings 性質

### ScaledownIdleTime ( 可選 , Integer )

定義沒有工作且Slurm節點終止的時間 (以分鐘為單位)。

預設值為 10。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### MungeKeySecretArn ( 可選 , String )

明文密碼 AWS Secrets Manager 碼的 Amazon 資源名稱 (ARN)，其中包含要在叢集中使用的 base64 編碼的密鑰。Slurm 這個 munge 金鑰將用來驗證用 Slurm 戶端命令和做為遠端伺服器的 Slurm 精靈之間的 RPC 呼叫。如果 MungeKeySecretArn 未提供，AWS ParallelCluster 將產生叢集的隨機 munge 金鑰。

#### Note

MungeKeySecretArn 從 AWS ParallelCluster 版本 3.8.0 開始支援。

#### Warning

如果 MungeKeySecretArn 是新增至現有叢集，則在復原或稍後移除時，ParallelCluster 將不會還原先前的「MungeKeySecretArn 金鑰」。相反，將生成一個新的隨機 munge 密鑰。

如果使 AWS ParallelCluster 用者具有該特定秘密資源的權限，MungeKeySecretArn 則會驗證。

[DescribeSecret](#) MungeKeySecretArn 是有效的，如果：

- 指定的密碼存在，並且
- 密碼是純文本，並包含一個有效的 base64 編碼字符串，並且
- 解碼的二進制 munge 密鑰的大小在 256 和 8192 位之間。

如果 pcluster 使用者 IAM 政策不包含 DescribeSecret，MungeKeySecretArn 則不會驗證並顯示警告訊息。如需詳細資訊，請參閱 [基本 AWS ParallelCluster pcluster 使用者原則](#)。

更新時 MungeKeySecretArn，必須停止運算叢集和所有登入節點。

如果在 ARN 保持不變的情況下修改秘密 ARN 中的密鑰值，則不會自動使用新的 munge 密鑰更新叢集。若要使用秘密 ARN 的新 munge 金鑰，您必須停止計算叢集和登入節點，然後從頭節點執行下列命令。

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

執行命令後，您可以同時恢復運算叢集和登入節點：新佈建的計算和登入節點將使用新的 munge 金鑰自動啟動。

要生成 base64 編碼的自定義蒙格密鑰，您可以使用與 munge 軟件一起分發的 [mungekey 實用程序](#)，然後使用操作系統中普遍可用的 base64 實用程序對其進行編碼。或者，您可以使用 bash（請在 32 和 1024 之間設置 bs 參數）

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

或 Python 下所示：

```
import random
import os
import base64

# key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

更新策略：已停止計算機群和登錄節點的新更新策略（錯誤地未在 3.7.0 中添加）。

### QueueUpdateStrategy（可選，String）

指定具有下列更新原則之 [SlurmQueues](#) 段落參數的取代策略：

[更新原則：必須停止運算叢集，或者 QueueUpdateStrategy 必須設定，才能變更此設定以進行更新。](#)

只有在叢集更新程序啟動時才會使用此 QueueUpdateStrategy 值。

有效值：COMPUTE\_FLEET\_STOP | DRAIN | TERMINATE

預設值：COMPUTE\_FLEET\_STOP

## DRAIN

已變更參數值之佇列中的節點設定為DRAINING。處於此狀態的節點不接受新工作，且執行中的工作會繼續完成。

節點變成 idle (DRAINED) 之後，如果節點是靜態的，則會取代節點，如果節點是動態的，則會終止節點。其他佇列中沒有變更參數值的其他節點不會受到影響。

此策略需要以變更的參數值取代所有佇列節點的時間，取決於執行中的工作負載。

## COMPUTE\_FLEET\_STOP

參數的預設QueueUpdateStrategy值。透過此設定，在執行叢集更新之前，您必須先[停止運算叢集](#)，才能更新[SlurmQueues](#)區段下的參數：

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

## TERMINATE

在參數值變更的佇列中，執行中的工作會終止，並立即關閉節點的電源。

靜態節點會被取代，並終止動態節點。

其他佇列中沒有變更參數值的其他節點不會受到影響。

[更新原則：更新期間不會分析此設定。](#)

### Note

QueueUpdateStrategy從 AWS ParallelCluster 版本 3.2.0 開始支援。

## EnableMemoryBasedScheduling ( 可選 , Boolean )

如果true在中Slurm啟用記憶體型排程。如需詳細資訊，請參閱[SlurmQueuesComputeResources//SchedulableMemory](#)。

預設值為 false。

### Warning

啟用記憶體型排程會影響排程Slurm器處理工作和節點配置的方式。

如需詳細資訊，請參閱 [Slurm記憶體式排程](#)。

**Note**

EnableMemoryBasedScheduling從 AWS ParallelCluster 版本 3.2.0 開始支援。

**Note**

從 AWS ParallelCluster 版本 3.7.0 開始，如果您在執行個體中設定多個[執行個體類型](#)，則EnableMemoryBasedScheduling可以啟用此功能。

適用於 AWS ParallelCluster 版本 3.2.0 至 3.6 版本。

**x**，EnableMemoryBasedScheduling如果您在執行個體中設定多個[執行個體類型](#)，則無法啟用。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### CustomSlurmSettings (可選, [Dict])

Slurm定義套用至整個叢集的自訂設定。

指定要附加到 AWS ParallelCluster 產生的slurm.conf檔案結尾的鍵值對的Slurm組態字典清單。

清單中的每個字典都會顯示為新增至Slurm組態檔案的單獨行。您可以指定簡單或複雜的參數。

簡單參數由單一 key pair 組成，如下列範例所示：

```
- Param1: 100  
- Param2: "SubParam1,SubParam2=SubValue2"
```

在Slurm配置中呈現的示例：

```
Param1=100  
Param2=SubParam1,SubParam2=SubValue2
```

複雜的Slurm配置參數由多個空格分隔的鍵值組成，如下面的例子所示：

```
- NodeName: test-nodes[1-10]
```

```
CPUs: 4
RealMemory: 4196
... # other node settings
- NodeSet: test-nodeset
Nodes: test-nodes[1-10]
... # other nodeset settings
- PartitionName: test-partition
Nodes: test-nodeset
... # other partition settings
```

範例，在Slurm模型組態中呈現：

```
NodeName=test-nodes[1-10] CPUs=4 RealMemory=4196 ... # other node settings
NodeSet=test-nodeset Nodes=test-nodes[1-10] ... # other nodeset settings
PartitionName=test-partition Nodes=test-nodeset ... # other partition settings
```

#### Note

自訂Slurm節點的名稱中不得包含 `-st-` 或 `-dy-` 模式。這些模式會保留給由管理的節點使用 AWS ParallelCluster。

如果您在中指定自訂Slurm組態參數 `CustomSlurmSettings`，則不得為指定自訂Slurm組態參數 `CustomSlurmSettingsIncludeFile`。

您只能指Slurm定未在中拒絕列出的組態參數。 `CustomSlurmSettings` 如需有關拒絕列出的Slurm組態參數的資訊，請參閱 [拒絕列出Slurm組態參數CustomSlurmSettings](#)

AWS ParallelCluster 僅檢查參數是否在拒絕清單中。AWS ParallelCluster 不會驗證您的自訂Slurm組態參數語法或語意。您必須負責驗證您的自訂Slurm組態參數。無效的自訂Slurm組態參數可能會造成Slurm協助程式失敗，進而導致叢集建立和更新失敗。

如需如何使用指定自訂Slurm組態參數的詳細資訊 AWS ParallelCluster，請參閱 [Slurm組態自訂](#)。

如需有關Slurm組態參數的詳細資訊，請參閱說明文件中的 [slurm.conf](#)。Slurm

[更新原則：您可以在更新期間變更此設定。](#)

#### Note

`CustomSlurmSettings` 從 AWS ParallelCluster 版本 3.6.0 開始支援。



## CustomSlurmSettingsIncludeFile ( 可選 , String )

Slurm定義套用至整個叢集的自訂設定。

指定自訂Slurm檔案，該檔案由要附加在產生的slurm.conf檔案結尾的自訂Slurm組態參數組AWS ParallelCluster 成。

您必須包含檔案的路徑。路徑可以以https://或開頭s3://。

如果您指定的自訂Slurm組態參數CustomSlurmSettingsIncludeFile，則不得指定的自訂Slurm組態參數CustomSlurmSettings。

### Note

自訂Slurm節點的名稱中不得包含-st-或-dy-模式。這些模式會保留給由管理的節點使用AWS ParallelCluster。

您只能指Slurm定未在中拒絕列出的組態參數。CustomSlurmSettingsIncludeFile如需有關拒絕列出的Slurm組態參數的資訊，請參閱 [拒絕列出Slurm組態參數CustomSlurmSettings](#)

AWS ParallelCluster 僅檢查參數是否在拒絕清單中。AWS ParallelCluster 不會驗證您的自訂Slurm組態參數語法或語意。您必須負責驗證您的自訂Slurm組態參數。無效的自訂Slurm組態參數可能會造成Slurm協助程式失敗，進而導致叢集建立和更新失敗。

如需如何使用指定自訂Slurm組態參數的詳細資訊 AWS ParallelCluster，請參閱[Slurm組態自訂](#)。

如需有關Slurm組態參數的詳細資訊，請參閱說明文件中的 [slurm.conf](#)。Slurm

[更新原則：您可以在更新期間變更此設定。](#)

### Note

CustomSlurmSettings從 AWS ParallelCluster 版本 3.6.0 開始支援。

## Database

(選擇性) 定義設定以在叢集上啟用「Slurm帳戶」。如需詳細資訊，請參閱 [Slurm會計與 AWS ParallelCluster](#)。

**Database:****Uri:** *string***UserName:** *string***PasswordSecretArn:** *string*

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

**Database 屬性****Uri ( 必填 , String )**

用作Slurm帳戶後端之資料庫伺服器的位址。此 URI 必須格式化為host:port且不得包含配置，例如mysql://。主機可以是 IP 位址，也可以是可由頭節點解析的 DNS 名稱。如果未提供連接埠，請 AWS ParallelCluster 使用MySQL預設連接埠 3306。

AWS ParallelCluster 將Slurm會計資料庫啟動至叢集，並且必須存取資料庫。

在發生下列情況之前，必須可存取資料庫：

- 隨即建立叢集。
- Slurm會計已透過叢集更新啟用。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

**UserName ( 必填 , String )**

Slurm用來連線至資料庫、寫入帳戶記錄及執行查詢的識別碼。使用者必須同時擁有資料庫的讀取和寫入權限。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

**PasswordSecretArn ( 必填 , String )**

包含UserName純文本密碼的 AWS Secrets Manager 密碼的 Amazon 資源名稱 ( ARN )。此密碼與UserName和Slurm帳戶一起使用，以便在資料庫伺服器上進行驗證。

**Note**

使用 AWS Secrets Manager 控制台創建密碼時，請務必選擇「其他類型的密碼」，選擇純文本，並且僅在密碼中包含密碼文本。

如需如何使用建立密碼 AWS Secrets Manager 的詳細資訊，請參閱[建立 AWS Secrets Manager 密碼](#)

如果使用者具有的權限 [DescribeSecret](#)，PasswordSecretArn則會驗證。PasswordSecretArn如果指定的密碼存在，則為有效。如果使用者 IAM 政策不包含DescribeSecret，則PasswordSecretArn不會驗證並顯示警告訊息。如需詳細資訊，請參閱 [基本 AWS ParallelCluster pcluster使用者原則](#)。

更新時PasswordSecretArn，必須停止運算叢集。如果密碼值變更，而秘密 ARN 沒有變更，叢集不會自動更新為新的資料庫密碼。若要更新新密碼值的叢集，您必須在運算叢集停止後，從頭節點內執行下列命令。

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

#### Warning

我們建議您僅在計算叢集停止時變更資料庫密碼，以避免遺失帳戶資料。

[更新原則：必須停止運算叢集，才能變更此設定以進行更新。](#)

### DatabaseName ( 可選, String )

要用於「Slurm會計」之資料庫伺服器 (由 Uri 參數定義) 上的資料庫名稱。

數據庫的名稱可以包含小寫字母，數字和下劃線。名稱不得超過 64 個字元。

此參數會對應至[思路密度 StorageLoc bd.conf](#) 的參數。

如果DatabaseName未提供，ParallelCluster 將使用叢集名稱來定義的值StorageLoc。

允許更新，但考量如下：DatabaseName

- 如果數據庫服務器上尚 DatabaseName 未存在具有名稱的數據庫，slurmdbd 將創建它。您有責任根據需要重新配置新的數據庫 (例如，添加會計實體-集群，帳戶，用戶，關聯，QoS 等)。
- 如果數據庫服務器上 DatabaseName 已經存在名稱的數據庫，slurmdbd 將使用它的會計功能。Slurm

[更新原則：必須停止運算叢集，才能變更此設定以進行更新。](#)

#### Note

Database從 3.3.0 版開始新增。

## ExternalSlurmdbd

(選擇性) 定義設定，以便透過外部 slurmdbd 伺服器啟用Slurm帳戶功能。如需詳細資訊，請參閱[使用的Slurm帳戶](#) AWS ParallelCluster。

### ExternalSlurmdbd:

Host: *string*

Port: *integer*

## ExternalSlurmdbd 屬性

### Host ( 必填 , String )

外部 slurmdbd 服務器的地址進行會計。Slurm主機可以是 IP 位址，也可以是可由頭節點解析的 DNS 名稱。

更新原則：您可以在更新期間變更此設定。

### Port ( 可選 , Integer )

服務所接聽的連接埠。預設值為 6819。

更新原則：您可以在更新期間變更此設定。

## Dns

(選擇性) 定義套用至整個叢集的設定。Slurm

### Dns:

DisableManagedDns: *boolean*

HostedZoneId: *string*

UseEc2Hostnames: *boolean*

## Dns 屬性

### DisableManagedDns ( 可選 , Boolean )

如果未建立叢集的 DNS 項目 true，且無法解析Slurm節點名稱。

依預設，AWS ParallelCluster 會建立 Route 53 託管區域，其中啟動時會註冊節點。預設值為 false。如果設定DisableManagedDns為true，則不會由建立託管區域 AWS ParallelCluster。

若要瞭解如何使用此設定在沒有網際網路存取權的子網路中部署叢集，請參閱[AWS ParallelCluster 在沒有互聯網訪問的單個子網中](#)。

#### Warning

叢集必須使用名稱解析系統才能正常運作。如果設定`DisableManagedDns`為`true`，您必須提供名稱解析系統。若要使用 Amazon EC2 預設 DNS，請`UseEc2Hostnames`將設定為`true`。或者，您也可以設定您自己的 DNS 解析程式，並確定在執行個體啟動時已註冊節點名稱。例如，您可以透過設定 [CustomActions/](#)來執行此操作[OnNodeStart](#)。

更新原則：如果變更此設定，則不允許更新。

### HostedZoneId ( 可選, String )

定義用於叢集 DNS 名稱解析的自訂 Route 53 託管區域識別碼。提供後，會在指定的託管區域中 AWS ParallelCluster 註冊叢集節點，而不會建立受管理的託管區域。

更新原則：如果變更此設定，則不允許更新。

### UseEc2Hostnames ( 可選, Boolean )

如果`true`，叢集運算節點設定為使用預設 EC2 主機名稱。也`SlurmNodeHostName`會使用此資訊更新。預設值為`false`。

若要瞭解如何使用此設定在沒有網際網路存取權的子網路中部署叢集，請參閱[AWS ParallelCluster 在沒有互聯網訪問的單個子網中](#)。

#### Note

此註釋與 AWS ParallelCluster 版本 3.3.0 開始無關。

對於 3.3.0 之前的 AWS ParallelCluster 支援版本：

當設定`UseEc2Hostnames`為`true`時，Slurm 組態檔案會使用 AWS ParallelCluster `prolog`和`epilog`指令碼設定：

- `prolog`執行以在分配每個作業時，將節點資訊新增至運算節點`/etc/hosts`上。
- `epilog`運行以清除寫入的內容`prolog`。

若要新增自訂`prolog`或`epilog`指令碼，請將它們分別新增至`/opt/slurm/etc/pcluster/prolog.d/`或`/opt/slurm/etc/pcluster/epilog.d/`資料夾。

更新原則：如果變更此設定，則不允許更新。

## SharedStorage 區段

(選擇性) 叢集的共用儲存裝置設定。

AWS ParallelCluster [支援使用 Amazon EBS、FSx 用於 ONTAP，以及用於 OpenZF 共享儲存磁碟區的 FSx、Amazon EFS 和 FSx for Lustre 共享儲存檔案系統](#) 或檔案快取。

在此 SharedStorage 區段中，您可以定義外部儲存或受管理的儲存體：

- 外部儲存是指您管理的現有磁碟區或檔案系統。AWS ParallelCluster 不會建立或刪除它。
- AWS ParallelCluster 受管理儲存是指建 AWS ParallelCluster 立且可以刪除的磁碟區或檔案系統。

如需 [共用儲存空間配額](#) 以及設定共用儲存空間的詳細資訊，請參閱使用 [共用儲存](#) 中的 AWS ParallelCluster。

### Note

如果 AWS Batch 作為排程器使用，則 FSx for Lustre 只能在叢集標頭節點上使用。

#### SharedStorage:

- `MountDir`: *string*  
`Name`: *string*  
`StorageType`: Ebs  
`EbsSettings`:  
`VolumeType`: *string*  
`Iops`: *integer*  
`Size`: *integer*  
`Encrypted`: *boolean*  
`KmsKeyId`: *string*  
`SnapshotId`: *string*  
`Throughput`: *integer*  
`VolumeId`: *string*  
`DeletionPolicy`: *string*  
`Raid`:  
`Type`: *string*  
`NumberOfVolumes`: *integer*
- `MountDir`: *string*  
`Name`: *string*  
`StorageType`: Efs  
`EfsSettings`:

```

Encrypted: boolean
KmsKeyId: string
EncryptionInTransit: boolean
IamAuthorization: boolean
PerformanceMode: string
ThroughputMode: string
ProvisionedThroughput: integer
FileSystemId: string
DeletionPolicy: string
- MountDir: string
  Name: string
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: integer
    DeploymentType: string
    ImportedFileChunkSize: integer
    DataCompressionType: string
    ExportPath: string
    ImportPath: string
    WeeklyMaintenanceStartTime: string
    AutomaticBackupRetentionDays: integer
    CopyTagsToBackups: boolean
    DailyAutomaticBackupStartTime: string
    PerUnitStorageThroughput: integer
    BackupId: string
    KmsKeyId: string
    FileSystemId: string
    AutoImportPolicy: string
    DriveCacheType: string
    StorageType: string
    DeletionPolicy: string
    DataRepositoryAssociations:
      - Name: string
        BatchImportMetaDataOnCreate: boolean
        DataRepositoryPath: string
        FileSystemPath: string
        ImportedFileChunkSize: integer
        AutoExportPolicy: string
        AutoImportPolicy: string
- MountDir: string
  Name: string
  StorageType: FsxOntap
  FsxOntapSettings:
    VolumeId: string

```

```

- MountDir: string
  Name: string
  StorageType: FsxOpenZfs
  FsxOpenZfsSettings:
    VolumeId: string
- MountDir: string
  Name: string
  StorageType: FileCache
  FileCacheSettings:
    FileCacheId: string

```

## SharedStorage 更新政策

- 對於受管理/外部 EBS、受管理 EFS 和受管理 FSx 資料庫，更新政策為 [更新原則：對於此清單值設定，必須停止運算叢集，或者QueueUpdateStrategy必須設定為新增值；移除現有值時必須停止運算叢集。](#)
- 對於外部 EFS、FSx 光碟、FSx ONTAP、FSx OpenZfs 和檔案快取，更新政策為：[更新原則：您可以在更新期間變更此設定。](#)

## SharedStorage 屬性

MountDir ( 必填, String )

掛接共用儲存裝置的路徑。

[更新原則：如果變更此設定，則不允許更新。](#)

Name ( 必填, String )

共用儲存裝置的名稱。您可以在更新設定時使用此名稱。

### Warning

如果您指定 AWS ParallelCluster 受管理的共用儲存區，並變更的值Name，則會刪除現有受管理的共用儲存區和資料，並建立新的受管理共用儲存區。Name使用叢集更新變更的值等同於以新的受管理共用儲存裝置取代現有的受管理共用儲存裝置。Name如果您需要保留現有共用儲存裝置中的資料，請務必在變更前備份資料。

[更新原則：對於此清單值設定，必須停止運算叢集，或者QueueUpdateStrategy必須設定為新增值；移除現有值時必須停止運算叢集。](#)



## StorageType ( 必填 , String )

共用儲存裝置的類型。支援的值為EbsEfs、FsxLustre、FsxOntap、和FsxOpenZfs。

如需詳細資訊，請參

閱[FsxLustreSettings](#)、[FsxOntapSettings](#)及[FsxOpenZfsSettings](#)。

### Note

如果您使用 AWS Batch 作為排程器，則 FSx for Lustre 只能在叢集標頭節點上使用。

更新原則：如果變更此設定，則不允許更新。

## EbsSettings

(選擇性) Amazon EBS 磁碟區的設定。

### EbsSettings:

VolumeType: *string*

Iops: *integer*

Size: *integer*

Encrypted: *boolean*

KmsKeyId: *string*

SnapshotId: *string*

VolumeId: *string*

Throughput: *integer*

DeletionPolicy: *string*

Raid:

Type: *string*

NumberOfVolumes: *integer*

更新原則：如果變更此設定，則不允許更新。

## EbsSettings 屬性

當設定[DeletionPolicy](#)為時Delete，如果刪除叢集或使用叢集更新移除磁碟區，則會刪除受管理磁碟區及其資料。

如需詳細資訊，請參閱使用[共用儲存](#)中的 AWS ParallelCluster。

## VolumeType ( 可選 , String )

指定 [Amazon EBS 磁碟區類型](#)。支援的值為gp2gp3、io1、io2、sc1、st1、和standard。預設值為 gp3。

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](#)。

更新原則：如果變更此設定，則不允許更新。

## Iops ( 可選 , Integer )

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VolumeType不同Size。

VolumeType = io1

預設值 Iops =

支援的值 Iops =

最大volume\_iops比volume\_size率 = 每個 GiB 的 50 IOPS。5000 IOPS 需要  
—volume\_size個至少 100 GiB。

VolumeType = io2

預設值 Iops =

支援的值 Iops = 100 至 64 萬 (對於io2區塊快速磁碟區為 256000) †

每Iops個 GiB 的最大Size比例 = 500 IOPS。5000 IOPS 需要至少 10 GiB 博Size的一個。

VolumeType = gp3

預設Iops值

支援的Iops值

每Iops個 GiB 的最大Size比例 = 500 IOPS。5000 IOPS 需要至少 10 GiB 博Size的一個。

† 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](#)，否則較早的io1磁碟區可能無法達到完整的效能。io2區塊快速磁碟區在R5b執行個體類型上支援高達 256000 的volume\_iops值。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](#)。

更新原則：您可以在更新期間變更此設定。

Size ( 可選 , Integer )

以吉位元組 (GiB) 為單位指定磁碟區大小。預設值為 35。

更新原則：如果變更此設定，則不允許更新。

Encrypted ( 可選 , Boolean )

指定磁碟區是否已加密。預設值為 true。

更新原則：如果變更此設定，則不允許更新。

KmsKeyId ( 可選 , String )

指定用於加密的自訂 AWS KMS 金鑰。此設定需要將 Encrypted 設定設定為 true。

更新原則：如果變更此設定，則不允許更新。

SnapshotId ( 可選 , String )


如果您使用快照做為磁碟區的來源，請指定 Amazon EBS 快照識別碼。

更新原則：如果變更此設定，則不允許更新。

VolumeId ( 可選 , String )

指定 Amazon EBS 磁碟區識別碼。針對 EbsSettings 例證指定此參數時，也只能指定 MountDir 參數。

磁碟區必須建立在相同的可用區域中 HeadNode。

 Note

在 AWS ParallelCluster 版本 3.4.0 中添加了多個可用區域。

更新原則：如果變更此設定，則不允許更新。

Throughput ( 可選 , Integer )

佈建磁碟區的輸送量 (以 MiB/s 為單位)，最大值為 1,000 Mb/s。

此設定只有在為時 VolumeType 才有效 gp3。支援的範圍是 125 到 1000，預設值為 125。

更新原則：您可以在更新期間變更此設定。

DeletionPolicy ( 可選 , String )

指定刪除叢集或移除磁碟區時，磁碟區是否應保留、刪除或快照。支援的值為DeleteRetain、和Snapshot。預設值為 Delete。

當DeletionPolicy設定為Delete，如果刪除叢集或使用叢集更新移除磁碟區，則會刪除受管理磁碟區及其資料。

如需詳細資訊，請參閱 [共用儲存](#)。

更新原則：您可以在更新期間變更此設定。

**Note**

DeletionPolicy從 AWS ParallelCluster 版本 3.2.0 開始支援。

## Raid

(選擇性) 定義 RAID 卷冊的組態。

Raid:

Type: *string*

NumberOfVolumes: *integer*

更新原則：如果變更此設定，則不允許更新。

### Raid 屬性

Type ( 必填 , String )

定義 RAID 陣列的類型。支援的值為「0」(條紋) 和「1」(鏡像)。

更新原則：如果變更此設定，則不允許更新。

NumberOfVolumes ( 可選 , Integer )

定義用來建立 RAID 陣列的 Amazon EBS 磁碟區數量。支援的值範圍為 2-5。預設值 (定義Raid設定時) 為 2。

更新原則：如果變更此設定，則不允許更新。

## EfsSettings

(選擇性) Amazon EFS 檔案系統的設定。

```
EfsSettings:  
  Encrypted: boolean  
  KmsKeyId: string  
  EncryptionInTransit: boolean  
  IamAuthorization: boolean  
  PerformanceMode: string  
  ThroughputMode: string  
  ProvisionedThroughput: integer  
  FileSystemId: string  
  DeletionPolicy: string
```

更新原則：如果變更此設定，則不允許更新。

### EfsSettings 屬性

當 [DeletionPolicy](#) 設定為 `Delete`，如果叢集遭刪除，或使用叢集更新移除檔案系統，則會刪除受管理的檔案系統及其資料。

如需詳細資訊，請參閱使用 [共用儲存](#) 中的 AWS ParallelCluster。

Encrypted ( 可選 , Boolean )

指定 Amazon EFS 檔案系統是否已加密。預設值為 `false`。

更新原則：如果變更此設定，則不允許更新。

KmsKeyId ( 可選 , String )

指定用於加密的自訂 AWS KMS 金鑰。此設定需要將 `Encrypted` 設定設定為 `true`。

更新原則：如果變更此設定，則不允許更新。

EncryptionInTransit ( 可選 , Boolean )

如果設定為 `true`，則會使用傳輸層安全性 (TLS) 掛接 Amazon EFS 檔案系統。依預設，此設定為 `false`。

#### Note

如果 AWS Batch 用作排程器，則 `EncryptionInTransit` 不支援。

**Note**

EncryptionInTransit 已從 AWS ParallelCluster 版本 3.4.0 開始新增。

更新原則：如果變更此設定，則不允許更新。

IamAuthorization ( 可選 , Boolean )

如果設定為 true , Amazon EFS 會使用系統的 IAM 身分進行驗證。依預設 , 此設定為 false。

**Note**

如果將 IamAuthorization 設定為 true , 則 EncryptionInTransit 也必須設定為 true。

**Note**

如果 AWS Batch 用作排程器 , 則 IamAuthorization 不支援。

**Note**

IamAuthorization 已從 AWS ParallelCluster 版本 3.4.0 開始新增。

更新原則：如果變更此設定，則不允許更新。

PerformanceMode ( 可選 , String )

指定 Amazon EFS 檔案系統的效能模式。支援的值為 generalPurpose 和 maxIO。預設值為 generalPurpose。如需詳細資訊 , 請參閱《Amazon Elastic File System 使用者指南》中的[效能模式](#)。

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。

更新原則：如果變更此設定，則不允許更新。

## ThroughputMode ( 可選 , String )

指定 Amazon EFS 檔案系統的輸送量模式。支援的值為 `bursting` 和 `provisioned`。預設值為 `bursting`。使 `provisioned` 用時，`ProvisionedThroughput` 必須指定。

更新原則：您可以在更新期間變更此設定。

## ProvisionedThroughput ( 當 `ThroughputMode` 是時需要 `provisioned` , Integer )

定義 Amazon EFS 檔案系統的佈建輸送量 (單位為 MIB/s)，以 MIB/s 為單位。這對應於 Amazon EFS API [ProvisionedThroughputInMibps](#) 參考中的參數。

如果使用此參數，則必須將 `ThroughputMode` 設為 `provisioned`。

支援的範圍為 1-1024。聯絡 AWS Support 以請求增加限制。

更新原則：您可以在更新期間變更此設定。

## FileSystemId ( 可選 , String )

定義現有檔案系統的 Amazon EFS 檔案系統 ID。

如果叢集設定為跨越多個可用區域，您必須在叢集使用的每個可用區域中定義檔案系統掛載目標。

指定此值時，只 `MountDir` 能指定。無法指 `EfsSettings` 定其他項目。

如果您設定此選項，則您定義的檔案系統必須符合下列條件：

- 檔案系統在每個叢集的可用區域中都有一個現有的掛載目標，並允許來自和的輸入 `HeadNode` 和輸出 NFS 流量 `ComputeNodes`。在 [排程//網路 `SlurmQueues`](#) 中設定了多個可用區域 [SubnetIds](#)。

若要確定叢集與檔案系統之間允許流量，您可以執行下列其中一項作業：

- 設定掛載目標的安全群組，以允許進出叢集子網路 CIDR 或前置碼清單的流量。

### Note

AWS ParallelCluster 驗證連接埠是否已開啟，以及是否已設定 CIDR 或前置詞清單。  
AWS ParallelCluster 不驗證 CIDR 塊或前綴列表的內容。

- 使用 [//SecurityGroups](#) 和 [SlurmQueuesNetworkingHeadNode](#) 來設定叢集節點的自訂安全性群組 [SecurityGroups](#)。 [Networking](#) 必須將自訂安全群組設定為允許叢集和檔案系統之間的流量。

**Note**

如果所有叢集節點都使用自訂安全群組，則 AWS ParallelCluster 僅驗證連接埠是否已開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

**Warning**

只 OneZone 有在所有運算節點和主節點都位於相同的可用區域時，才支援 EFS。EFS 只 OneZone 能有一個裝載目標。

**Note**

在 AWS ParallelCluster 版本 3.4.0 中添加了多個可用區域。

更新原則：如果變更此設定，則不允許更新。

DeletionPolicy ( 可選, String )

指定當檔案系統從叢集中移除或刪除叢集時，是否應保留或刪除檔案系統。支援的值為 Delete 和 Retain。預設值為 Delete。

當設定 [DeletionPolicy](#) 為 Delete，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除受管理的檔案系統及其資料。

如需詳細資訊，請參閱 [共用儲存](#)。

更新原則：您可以在更新期間變更此設定。

**Note**

DeletionPolicy 從 AWS ParallelCluster 版本 3.3.0 開始支援。



## FsxLustreSettings

### Note

您必須定義FsxLustre是FsxLustreSettings否為指定[StorageType](#)。

(選擇性) Lustre 檔案系統之 FSx 的設定。

#### FsxLustreSettings:

```
StorageCapacity: integer  
DeploymentType: string  
ImportedFileChunkSize: integer  
DataCompressionType: string  
ExportPath: string  
ImportPath: string  
WeeklyMaintenanceStartTime: string  
AutomaticBackupRetentionDays: integer  
CopyTagsToBackups: boolean  
DailyAutomaticBackupStartTime: string  
PerUnitStorageThroughput: integer  
BackupId: string # BackupId cannot coexist with some of the fields  
KmsKeyId: string  
FileSystemId: string # FileSystemId cannot coexist with other fields  
AutoImportPolicy: string  
DriveCacheType: string  
StorageType: string  
DeletionPolicy: string
```

更新原則：如果變更此設定，則不允許更新。

### Note

如果 AWS Batch 作為排程器使用，則 FSx for Lustre 只能在叢集標頭節點上使用。

## FsxLustreSettings 屬性

當設定[DeletionPolicy](#)為時Delete，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除受管理的檔案系統及其資料。

如需詳細資訊，請參閱 [共用儲存](#)。

## StorageCapacity ( 必填 , Integer )

設定 Lustre 檔案系統的 FSx 儲存容量 (以 GiB 為單位)。StorageCapacity如果您要建立新的檔案系統，則為必要項目。StorageCapacity如果FileSystemId已指定BackupId或，請勿包含。

- 對於SCRATCH\_2PERSISTENT\_1、和PERSISTENT\_2部署類型，有效值為 1200 GiB、2400 GiB 和 2400 GiB 的增量。
- 對於 SCRATCH\_1 部署類型，有效值為 1200 GiB、2400 GiB，以及以 3600 GiB 遞增的倍數。

更新原則：如果變更此設定，則不允許更新。

## DeploymentType ( 可選 , String )

指定 Lustre 檔案系統之 FSx 的部署類型。支援的值為 SCRATCH\_1、SCRATCH\_2、PERSISTENT\_1 和 PERSISTENT\_2。預設值為 SCRATCH\_2。

當您需要臨時儲存SCRATCH\_1和較短的資料處理時，選擇和SCRATCH\_2部署類型。部SCRATCH\_2署類型提供資料的傳輸加密，以及高於SCRATCH\_1。

為長期儲存選擇PERSISTENT\_1部署類型，以及不敏感延遲的輸送量集中工作負載。PERSISTENT\_1支持加密傳輸中的數據。它可用於所有 FS AWS 區域 x for Lustre 可用。

為長期儲存以及需要最高等級 IOPS 和輸送量的延遲敏感工作負載選擇PERSISTENT\_2部署類型。PERSISTENT\_2支持固態硬盤存儲，並提供更高的存儲PerUnitStorageThroughput ( 高達 1000 MB/S /TIB )。 PERSISTENT\_2提供的數量有限 AWS 區域。如需有關部署類型和可用 AWS 區域 位置PERSISTENT\_2清單的詳細資訊，請參閱 Amazon FSx for Lustre 使用者指南中的適用於 Lustre 的 FSx [檔案系統部署選項](#)。

當您從支援此功能的 [Amazon EC2 執行個體存取SCRATCH\\_2或PERSISTENT\\_2部署類型檔案系統時PERSISTENT\\_1](#)，傳輸中的資料加密會自動啟用。

從支援的執行個體類型存取時 SCRATCH\_2PERSISTENT\_1，支援加密、和PERSISTENT\_2部署類型的傳輸中資料 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](#)。

### Note

AWS ParallelCluster 版本 3.2.0 增加了對PERSISTENT\_2部署類型的 Support。

更新原則：如果變更此設定，則不允許更新。

## ImportedFileChunkSize ( 可選 , Integer )

對於從資料儲存庫匯入的檔案，此值決定儲存在單一實體磁碟上的每個檔案 (以 MiB 為單位) 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。

預設區塊大小是 1,024 MiB (1 GiB)，最大為 512,000 MiB (500 GiB)。Amazon S3 物件大小的上限為 5 TB。

### Note

使用 PERSISTENT\_2 部署類型的檔案系統不支援此參數。如需有關如何設定資料儲存庫關聯的指示，請參閱 [Amazon FSx for Lustre 使用者指南中的將檔案系統連結至 S3 儲存貯體](#)。

更新原則：如果變更此設定，則不允許更新。

## DataCompressionType ( 可選 , String )

為 Lustre 檔案系統設定 FSx 的資料壓縮組態。支援的值為 LZ4。LZ4 指示使用 LZ4 演算法開啟資料壓縮。DataCompressionType 如果未指定，則在建立檔案系統時會關閉資料壓縮。

如需詳細資訊，請參閱 [Lustre 資料壓縮](#)。

更新原則：您可以在更新期間變更此設定。

## ExportPath ( 可選 , String )

Amazon S3 中的路徑，其中匯出 FSx for Lustre 的根目錄。只有在指定 ImportPath 參數時才支援此設定。路徑必須使用與中指定相同的 Amazon S3 儲存貯體 ImportPath。您可以提供選擇性字首，以便從 FSx for Lustre 檔案系統匯出新資料與變更的資料。如果未提供 ExportPath 值，則 FSx for Lustre 設定預設匯出路徑。s3://import-bucket/FSxLustre[creation-timestamp] 時間戳記使用 UTC 格式，例如 s3://import-bucket/FSxLustre20181105T222312Z。

Amazon S3 匯出儲存貯體和 ImportPath 指定的匯入儲存貯體必須相同。如果您僅指定儲存貯體名稱 (例如 s3://import-bucket)，則會取得檔案系統物件到 Amazon S3 儲存貯體物件的 1:1 對應。此對應表示 Amazon S3 中的輸入資料會在匯出時遭到覆寫。如果您在匯出路徑中提供自訂前置詞 (例如 s3://import-bucket/[custom-optional-prefix]，FSx for Lustre) 會將檔案系統的內容匯出到 Amazon S3 儲存貯體中的該匯出前綴。

**Note**

使用PERSISTENT\_2部署類型的檔案系統不支援此參數。按照 Amazon FSx for Lustre 使用者指南中的[將檔案系統連結到 S3 儲存貯體](#)中所述設定資料儲存庫關聯。

更新原則：如果變更此設定，則不允許更新。

ImportPath ( 可選 , String )

您用來做為 Lustre 檔案系統 FSx 的資料儲存庫的 Amazon S3 儲存貯體路徑 (包括選用的前置詞)。FSx for Lustre 檔案系統的根目錄會映射到您選取的 Amazon S3 儲存貯體根目錄。例如，s3://import-bucket/optional-prefix。如果您在 Amazon S3 儲存貯體名稱後面指定前綴，檔案系統只會載入具有該前綴的物件索引鍵。

**Note**

使用PERSISTENT\_2部署類型的檔案系統不支援此參數。按照 Amazon FSx for Lustre 使用者指南中的[將檔案系統連結到 S3 儲存貯體](#)中所述設定資料儲存庫關聯。

更新原則：如果變更此設定，則不允許更新。

WeeklyMaintenanceStartTime ( 可選 , String )

執行每週維護的首選開始時間。它的格"d:HH:MM"式為 UTC+0 時區。對於此格式，d是從 1 到 7 的工作日數字，從星期一開始，以星期日結束。此欄位需要引號。

更新原則：您可以在更新期間變更此設定。

AutomaticBackupRetentionDays ( 可選 , Integer )

自動備份的保留天數。將此設定為 0 會停用自動備份。支持的範圍是 0-90。預設值為 0。此設定僅適用於PERSISTENT\_1和PERSISTENT\_2部署類型。如需詳細資訊，請參閱[Amazon FSx for Lustre 使用者指南中的使用備份](#)。

更新原則：您可以在更新期間變更此設定。

CopyTagsToBackups ( 可選 , Boolean )

如果是true，請將 Lustre 檔案系統 FSx 的標籤複製到備份中。此值預設為 false。如設為 true，則所有檔案系統標籤都會複製到使用者不指定標籤的所有自動和使用者啟動的備份。如果此值為 true，而您指定一或多個標籤，只有指定的標籤會複製到備份。如果您在建立使用者啟動

的備份時指定一或多個標籤，則無論此值為何，都不會從檔案系統複製任何標籤。此設定僅適用於PERSISTENT\_1和PERSISTENT\_2部署類型。

更新原則：如果變更此設定，則不允許更新。

DailyAutomaticBackupStartTime ( 可選 , String )

每日週期性時間，格HH:MM式為。HH是一天中的零填充小時 (00-23)。MM是小時中的零填充分鐘 (00-59)。例如，05:00指定每天上午 5 點。此設定僅適用於PERSISTENT\_1和PERSISTENT\_2部署類型。

更新原則：您可以在更新期間變更此設定。

PerUnitStorageThroughput(部署類型PERSISTENT\_1和PERSISTENT\_2部署類型是必要的Integer)

描述每個 1 TB 儲存體的讀取和寫入輸送量，以 MB/S/TiB 為單位。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 PerUnitStorageThroughput (MB/s/TiB)。對於 2.4 TiB 的檔案系統，佈建 50 MB/s/TiB 的 PerUnitStorageThroughput 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。

有效值：

永久性\_1 固態硬碟儲存裝置：50、100、200 MBTiB 秒。

永久性\_1 個硬碟儲存空間：12、40 MB /TiB。

持久性\_2 固態硬碟儲存裝置：125、250、500、1000 MBTiB 次。

更新原則：如果變更此設定，則不允許更新。

BackupId ( 可選 , String )

指定用於從現有備份還原 Lustre 檔案系統 FSx 的備份識別碼。指定BackupId設定後 AutoImportPolicyDeploymentType，不得指定ExportPathKmsKeyIdImportPath、ImportedFileChunkSize、StorageCapacity、、、和PerUnitStorageThroughput設定。這些設定會從備份中讀取。此外 AutoImportPolicyExportPath，不得指定ImportPath、、、和ImportedFileChunkSize設定。這對應於[BackupId](#)屬性。

更新原則：如果變更此設定，則不允許更新。

KmsKeyId ( 可選 , String )

用來為 Lustre 檔案系統的永久 FSx 資料加密 FSx 的 AWS Key Management Service (AWS KMS) 金鑰識別碼，用來為靜態檔案系統的 FSx 加密。如果未指定，則會使用 Lustre 管理金鑰的

FSx。Lustre 檔案系統的SCRATCH\_1和 SCRATCH\_2 FSx 一律會使用 FSx for Lustre 管理的金鑰進行靜態加密。如需詳細資訊，請參閱 AWS Key Management Service API 參考中的[加密](#)。

更新原則：如果變更此設定，則不允許更新。

FileSystemId ( 可選 , String )

指定 Lustre 檔案系統之現有 FSx 的識別碼。

如果指定此選項，則僅會使用中FsxLustreSettings的MountDir和FileSystemId設定。將忽略中的FsxLustreSettings所有其他設定。

**Note**

如果使用 AWS Batch 排程器，則 FSx for Lustre 只能在頭節點上使用。

**Note**

檔案系統必須與允許透過連接埠 988、1021、1022 和 1023 之輸入和輸出 TCP 流量的安全性群組相關聯。

請執行下列其中一項動作，確定叢集與檔案系統之間允許流量：

- 設定檔案系統的安全性群組，以允許進出叢集子網路 CIDR 或前置碼清單的流量。

**Note**

AWS ParallelCluster 驗證連接埠是否已開啟，以及是否已設定 CIDR 或前置詞清單。  
AWS ParallelCluster 不驗證 CIDR 塊或前綴列表的內容。

- 使用//[SecurityGroups](#)和 [SlurmQueuesNetworkingHeadNode](#)/來設定叢集節點的自訂安全性群組[SecurityGroups](#)。[Networking](#)必須將自訂安全群組設定為允許叢集和檔案系統之間的流量。

**Note**

如果所有叢集節點都使用自訂安全群組，則 AWS ParallelCluster 僅驗證連接埠是否已開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新原則：如果變更此設定，則不允許更新。

AutoImportPolicy ( 可選 , String )

當您建立 FSx for Lustre 檔案系統時，您現有的 Amazon S3 物件會顯示為檔案和目錄清單。使用此屬性可選擇 FSx for Lustre 在連結的 Amazon S3 儲存貯體中新增或修改物件時，如何將檔案和目錄清單保持在最新狀態。AutoImportPolicy 可以具有以下值：

- NEW-自動匯入已開啟。FSx for Lustre 會自動匯入新增至連結 Amazon S3 儲存貯體的任何新物件的目錄清單，而這些物件目前不存在於 Lustre 檔案系統的 FSx 中。
- NEW\_CHANGED-自動匯入已開啟。FSx for Lustre 會在您選擇此選項後，自動匯入新增至 Amazon S3 儲存貯體的任何新物件的檔案和目錄清單，以及 Amazon S3 儲存貯體中已變更的任何現有物件。
- NEW\_CHANGED\_DELETED-自動匯入已開啟。FSx for Lustre 會自動匯入新增至 Amazon S3 儲存貯體的任何新物件的檔案和目錄清單、Amazon S3 儲存貯體中已變更的任何現有物件，以及選擇此選項後，在 Amazon S3 儲存貯體中刪除的任何物件。

**Note**

在 3.1.1 AWS ParallelCluster 版本中添加了 Support。NEW\_CHANGED\_DELETED

如果AutoImportPolicy未指定，則會關閉自動匯入。FSx for Lustre 只會在建立檔案系統時，從連結的 Amazon S3 儲存貯體更新檔案和目錄清單。選擇此選項後，FSx for Lustre 不會更新任何新物件或變更物件的檔案和目錄清單。

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](#)。

**Note**

使用PERSISTENT\_2部署類型的檔案系統不支援此參數。如需有關如何設定資料儲存庫關聯的指示，請參閱 [Amazon FSx for Lustre 使用者指南中的將檔案系統連結至 S3 儲存貯體](#)。

更新原則：如果變更此設定，則不允許更新。

DriveCacheType ( 可選 , String )

指定檔案系統具有 SSD 磁碟機快取。只有在設定設定為且StorageType設定為時HDD，才能DeploymentType設定此選項PERSISTENT\_1。這對應於DriveCacheType屬性。如需詳細資訊，請參閱《亞馬遜 FSx for Lustre 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 DriveCacheType 設定。

更新原則：如果變更此設定，則不允許更新。

StorageType ( 可選 , String )

為您建立的 Lustre 檔案系統設定 FSx 的儲存類型。有效值為 SSD 和 HDD。

- 設為 SSD 以使用固態硬碟儲存體。
- 設定為 HDD 使用硬碟儲存空間。HDDPERSISTENT 部署類型支援。

預設值為 SSD。如需詳細資訊，請參閱視窗版 Amazon FSx 使用者指南中的 [儲存類型選項](#) 和 [Amazon FSx 使用者指南中的多個儲存選項](#)。

更新原則：如果變更此設定，則不允許更新。


DeletionPolicy ( 可選 , String )

指定當檔案系統從叢集中移除或刪除叢集時，是否應保留或刪除檔案系統。支援的值為 Delete 和 Retain。預設值為 Delete。

當設定 [DeletionPolicy](#) 為 Delete，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除受管理的檔案系統及其資料。

如需詳細資訊，請參閱 [共用儲存](#)。

更新原則：您可以在更新期間變更此設定。

 Note

DeletionPolicy 從 AWS ParallelCluster 版本 3.3.0 開始支援。

DataRepositoryAssociations ( 可選 , String )

DRA 清單 (每個檔案系統最多 8 個)

每個資料儲存庫關聯必須具有唯一的 Amazon FSx 檔案系統目錄，以及與其相關聯的唯一 S3 儲存貯體或字首。

您無法 [ImportPath](#) 在使用 DRA 的同時使用 [ExportPath](#) 和 [FsxLustreSettings](#)



更新原則：您可以在更新期間變更此設定。

Name ( 必填 , String )

DRA 的名稱。您可以在更新設定時使用此名稱。

更新原則：如果變更此設定，則不允許更新。

BatchImportMetaDataOnCreate ( 可選 , Boolean )

布林值標記，指出要匯入中繼資料的匯入資料儲存庫任務是否應在建立資料儲存庫關聯之後執行。如果此標記設定為 true，任務便會執行。

預設值：false

更新原則：如果變更此設定，則不允許更新。

DataRepositoryPath ( 必填 , String )

將連結至檔案系統之 Amazon S3 資料儲存庫的路徑。路徑可以是 S3 儲存貯體或字首，格式為 s3://myBucket/myPrefix/。此路徑指定 S3 資料儲存庫檔案匯出或匯入的位置。

不能與其他 DRA 重疊

模式：`^[^\u0000\u0085\u2028\u2029\r\n]{3,4357}$`

下限：3

上限：4357

更新原則：如果變更此設定，則不允許更新。

FileSystemPath ( 必填 , String )

Amazon FSx for Lustre 檔案系統上指向高階目錄 (如 /ns1/) 或子目錄 (如 /ns1/subdir/) 的路徑，此路徑會透過 DataRepositoryPath 完整映射。名稱需要以正斜線開頭。兩個資料儲存庫關聯的檔案系統路徑不可重疊。例如，如果某個資料儲存庫已經與檔案系統路徑 /ns1/ 建立關聯，則您無法將其他資料儲存庫連結到檔案系統路徑 /ns1/ns2。

此路徑指定檔案系統匯出或匯入檔案的位置。此檔案系統目錄只能連結到一個 Amazon S3 儲存貯體，其他 S3 儲存貯體無法連結至該目錄。

不能與其他 DRA 重疊

**Note**

如果您僅指定一個正斜線 (/) 作為檔案系統路徑，則只能將一個資料儲存庫連結至檔案系統。您只能指定 "/" 作為與檔案系統相關聯之第一個資料儲存庫的檔案系統路徑。

模式：`^[^\u0000\u0085\u2028\u2029\r\n]{1,4096}$`

下限：1

上限：4096

更新原則：如果變更此設定，則不允許更新。

ImportedFileChunkSize ( 可選 , Integer )

如果是從資料儲存庫匯入資料，此值會規定單一實體磁碟上存放的每個檔案的分割計數和資料量上限 (MiB)。單一檔案分割分佈所在的磁碟數上限會受組成檔案系統或快取的磁碟總數所限。

預設區塊大小是 1,024 MiB (1 GiB)，最大為 512,000 MiB (500 GiB)。Amazon S3 物件大小的上限為 5 TB。

下限：1

上限：4096

更新原則：您可以在更新期間變更此設定。

AutoExportPolicy ( 可選 , Array of strings )

清單可以包含下列一或多個值：

- NEW：將新檔案和目錄新增至檔案系統時，會將新檔案和目錄自動匯出至資料儲存庫。
- CHANGED：對檔案系統上檔案和目錄所做的變更會自動匯出至資料儲存庫。
- DELETED：在檔案系統上刪除檔案和目錄時，會自動刪除資料儲存庫中的檔案和目錄。

您可以為 AutoExportPolicy 定義任何事件類型組合。

上限：3

更新原則：您可以在更新期間變更此設定。

AutoImportPolicy ( 可選 , Array of strings )

清單可以包含下列一或多個值：

- NEW : Amazon FSx 會自動匯入任何新增至連結 S3 儲存貯體且目前不存在於 FSx 檔案系統中的檔案中繼資料。
- CHANGED : Amazon FSx 會在資料儲存庫中的檔案變更時，自動更新檔案中繼資料，並使檔案系統上的現有檔案內容失效。
- DELETED : Amazon FSx 會在檔案從資料儲存庫中刪除時自動刪除檔案系統上的對應檔案。

您可以為 AutoImportPolicy 定義任何事件類型組合。

上限：3

[更新原則：您可以在更新期間變更此設定。](#)

## FsxOntapSettings

### Note

您必須定義 FsxOntap 是 FsxOntapSettings 否為指定 [StorageType](#)。

(選擇性) 適用於 ONTAP 檔案系統的 FSx 設定。

[FsxOntapSettings](#):  
[VolumeId](#): *string*

## FsxOntapSettings 屬性

VolumeId ( 必填 , String )

指定 ONTAP 系統之現有 FSx 的磁碟區識別碼。

### Note

- 如果使用 AWS Batch 排程器，則僅在頭節點上可用於 ONTAP 的 FSx。
- 如果 ONTAP 部署類型的 FSx 是 Multi-AZ，請確定頭節點子網路的路由表已正確設定。
- 在 3.2.0 AWS ParallelCluster 版中新增了對 FSx 的 Support 援。
- 檔案系統必須與允許透過連接埠 111、635、2049 和 4046 之輸入和輸出 TCP 和 UDP 流量的安全性群組相關聯。

請執行下列其中一個動作，確定叢集與檔案系統之間允許流量：

- 設定檔案系統的安全性群組，以允許進出叢集子網路 CIDR 或前置碼清單的流量。

**Note**

AWS ParallelCluster 驗證連接埠是否已開啟，以及是否已設定 CIDR 或前置詞清單。AWS ParallelCluster 不驗證 CIDR 塊或前綴列表的內容。

- 使用//[SecurityGroups](#)和 [SlurmQueuesNetworkingHeadNode](#)/來設定叢集節點的自訂安全性群組[SecurityGroups](#)。 [Networking](#)必須將自訂安全群組設定為允許叢集和檔案系統之間的流量。

**Note**

如果所有叢集節點都使用自訂安全群組，則 AWS ParallelCluster 僅驗證連接埠是否已開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新原則：如果變更此設定，則不允許更新。

## FsxOpenZfsSettings

**Note**

您必須定義FsxOpenZfs是FsxOpenZfsSettings否為指定[StorageType](#)。

(選擇性) 適用於 OpenZFS 檔案系統之 FSx 的設定。

[FsxOpenZfsSettings](#):

[VolumeId](#): *string*

更新原則：如果變更此設定，則不允許更新。

### FsxOpenZfsSettings 屬性

VolumeId ( 必填 , String )

指定 OpenZFS 系統之現有 FSX 的磁碟區識別碼。

**Note**

- 如果使用 AWS Batch 排程器，OpenZF 的 FSx 只能在標頭節點上使用。
- 在 AWS ParallelCluster 3.2.0 版中新增了對 FSx 的 Support 援。
- 檔案系統必須與允許透過連接埠 111、2049、20001、20002 和 20003 的輸入和輸出 TCP 和 UDP 流量相關聯的安全性群組。

請執行下列其中一項動作，確定叢集與檔案系統之間允許流量：

- 設定檔案系統的安全性群組，以允許進出叢集子網路 CIDR 或前置碼清單的流量。

**Note**

AWS ParallelCluster 驗證連接埠是否已開啟，以及是否已設定 CIDR 或前置詞清單。AWS ParallelCluster 不驗證 CIDR 塊或前綴列表的內容。

- 使用//[SecurityGroups](#)和 [SlurmQueuesNetworkingHeadNode](#)/來設定叢集節點的自訂安全性群組 [SecurityGroups](#)。 [Networking](#) 必須將自訂安全群組設定為允許叢集和檔案系統之間的流量。

**Note**

如果所有叢集節點都使用自訂安全群組，則 AWS ParallelCluster 僅驗證連接埠是否已開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新原則：如果變更此設定，則不允許更新。

## FileCacheSettings

**Note**

您必須定義 FileCache 是 FileCacheSettings 否為指定 [StorageType](#)。

(選擇性) 「檔案快取」的設定。

[FileCacheSettings](#):

`FileCacheId`: *string*

更新原則：如果變更此設定，則不允許更新。

## FileCacheSettings 屬性

FileCacheId ( 必填, String )

指定現有檔案快取的檔案快取 ID。

### Note

- 檔案快取不支援 AWS Batch 排程器。
- 3.7.0 AWS ParallelCluster 版中增加了對文件緩存的 Support。
- 檔案系統必須與允許透過連接埠 988 輸入和輸出 TCP 流量的安全性群組相關聯。

請執行下列其中一項動作，確定叢集與檔案系統之間允許流量：

- 設定檔案快取的安全性群組，以允許進出叢集子網路 CIDR 或前置碼清單的流量。

### Note

AWS ParallelCluster 驗證連接埠是否已開啟，以及是否已設定 CIDR 或前置詞清單。AWS ParallelCluster 不驗證 CIDR 塊或前綴列表的內容。

- 使用//[SecurityGroups](#)和 [SlurmQueuesNetworkingHeadNode](#)/來設定叢集節點的自訂安全性群組 [SecurityGroups](#)。 [Networking](#) 必須將自訂安全群組設定為允許叢集和檔案系統之間的流量。

### Note

如果所有叢集節點都使用自訂安全群組，則 AWS ParallelCluster 僅驗證連接埠是否已開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新原則：如果變更此設定，則不允許更新。

## Iam 區段

(選擇性) 指定叢集的 IAM 屬性。

```
Iam:  
  Roles:  
    LambdaFunctionsRole: string  
    PermissionsBoundary: string  
    ResourcePrefix: string
```

[更新原則：您可以在更新期間變更此設定。](#)

### Iam 屬性

PermissionsBoundary (可選, String)

IAM 政策的 ARN，用作由 AWS ParallelCluster 建立的所有角色的許可界限。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 實體許可界限](#)。格式是 `arn:${Partition}:iam::${Account}:policy/${PolicyName}`。

[更新原則：您可以在更新期間變更此設定。](#)

Roles(選擇性)

指定叢集所使用之 IAM 角色的設定。

[更新原則：您可以在更新期間變更此設定。](#)

LambdaFunctionsRole (可選, String)

要用於 AWS Lambda 的 IAM 角色的 ARN。這會覆寫附加至支援 AWS CloudFormation 自訂資源的所有 Lambda 函數的預設角色。Lambda 需要設定為允許擔任該角色的主體。這不會覆寫使用的 Lambda 函數的角色 AWS Batch。格式是 `arn:${Partition}:iam::${Account}:role/${RoleName}`。

[更新原則：您可以在更新期間變更此設定。](#)

ResourcePrefix(選擇性)

為建立的 IAM 資源指定路徑或名稱前置詞 AWS ParallelCluster。

資源前綴必須遵循 [IAM 指定的命名規則](#)：

- 一個名稱最多可包含 30 個字元。
- 名稱只能是不含斜線 (/) 字元的字串。
- 路徑最多可以有 512 個字元。
- 路徑必須以斜線 (/) 開頭和結尾。它可以包含多個斜杠 (/) 開始和結束斜杠 (/) 之間。
- 您可以結合路徑和名稱/path/name。

指定名稱。

```
Iam:  
  ResourcePrefix: my-prefix
```

指定路徑。

```
Iam:  
  ResourcePrefix: /org/dept/team/project/user/
```

指定路徑和名稱。

```
Iam:  
  ResourcePrefix: /org/dept/team/project/user/my-prefix
```

如果您指定/my-prefix，則會傳回錯誤。

```
Iam:  
  ResourcePrefix: /my-prefix
```

返回配置錯誤。路徑必須有兩個 / s。前綴本身不能有 / s。

[更新原則：如果變更此設定，則不允許更新。](#)

## LoginNodes 區段

### Note

在 AWS ParallelCluster 版本 3.7.0 中增加了對LoginNodes的 Support。



(選擇性) 指定登入節點集區的組態。

#### LoginNodes:

##### Pools:

- Name: *string*
- Count: *integer*
- InstanceType: *string*
- GracetimePeriod: *integer*
- Image:
  - CustomAmi: *string*
- Ssh:
  - KeyName: *string*
- Networking:
  - SubnetIds:
    - *string*
  - SecurityGroups:
    - *string*
  - AdditionalSecurityGroups:
    - *string*
- Iam:
  - InstanceRole: *string*
  - InstanceProfile: *string*
  - AdditionalIamPolicies:
    - Policy: *string*

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

## LoginNodes 屬性

## Pools 屬性

定義具有相同資源配置的登入節點群組。只能指定單一集區。

#### Pools:

- Name: *string*
- Count: *integer*
- InstanceType: *string*
- GracetimePeriod: *integer*
- Image:
  - CustomAmi: *string*
- Ssh:
  - KeyName: *string*
- Networking:

```
SubnetIds:
  - string
SecurityGroups:
  - string
AdditionalSecurityGroups:
  - string
Iam:
  InstanceRole: string
  InstanceProfile: string
  AdditionalIamPolicies:
    - Policy: string
```

### Name(必填String)

指定LoginNodes集區的名稱。這是用來標記資LoginNodes源。

更新原則：如果變更此設定，則不允許更新。

### Count(必填Integer)

指定要保持作用中的登入節點數目。

更新原則：您可以在更新期間變更此設定。

### InstanceType(必填String)

指定用於登入節點的 Amazon EC2 執行個體類型。執行個體類型的架構必須與用於 SlurmInstanceType設定的架構相同。

更新政策：如果登入節點集區停止，則可以變更此設定。

### GracetimePeriod(選擇性Integer)

指定通知已登入使用者登入節點將要解除委任與實際停止事件之間經過的最小時間 (以分鐘為單位)。的有效值為 GracetimePeriod 3 至 120 分鐘。預設值為 60 分鐘。

#### Note

觸發事件涉及多個 AWS 服務之間的交互。有時，網路延遲和資訊傳播可能需要一些時間，因此寬限期可能會因為 AWS 服務的內部延遲而需要比預期更長的時間。

更新原則：您可以在更新期間變更此設定。

## Image(選擇性)

定義登入節點的映像配置。

```
Image:  
CustomAmi: String
```

### CustomAmi(選擇性String)

指定用於佈建登入節點的自訂 AMI。如果未指定，則預設值為中指定的值[HeadNode 區段](#)。

更新原則：如果變更此設定，則不允許更新。

## Ssh(選擇性)

定義登入節點的ssh組態。

```
Ssh:  
KeyName: string
```

### KeyName(選擇性String)

指定用來登入登入節點的ssh金鑰。如果未指定，則預設值為中指定的值[HeadNode 區段](#)。

更新原則：如果變更此設定，則不允許更新。

## Networking(必填)

```
Networking:  
SubnetIds:  
- string  
SecurityGroups:  
- string  
AdditionalSecurityGroups:  
- string
```

### SubnetIds(必填[String])

佈建登入節點集區的現有子網路識別碼。您只能定義一個子網路。

更新原則：如果變更此設定，則不允許更新。

## SecurityGroups(選擇性[String])

用於登入節點集區的安全群組清單。如果未指定安全性群組，請為您 AWS ParallelCluster 建立安全性群組。

更新原則：如果變更此設定，則不允許更新。

## AdditionalSecurityGroups(選擇性[String])

用於登入節點集區的其他安全群組清單。

更新原則：如果變更此設定，則不允許更新。

## Iam(選擇性)

指定要在登入節點上使用的執行個體角色或執行個體設定檔，以覆寫叢集的預設執行個體角色或執行個體設定檔。

Iam:

```
InstanceRole: string  
InstanceProfile: string  
AdditionalIamPolicies:  
- Policy: string
```

## InstanceProfile(選擇性String)

指定要覆寫預設登入節點執行個體設定檔的執行個體設定檔。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`。如果已指定，則無法指AdditionalIamPolicies定InstanceRole和設定。

更新原則：如果變更此設定，則不允許更新。

## InstanceRole(選擇性String)

指定要覆寫預設登入節點執行個體角色的執行個體角色。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:role/RoleName`。如果已指定，則無法指AdditionalIamPolicies定S3Access和設定。如果已指定，則無法指AdditionalIamPolicies定InstanceProfile和設定。

更新原則：如果變更此設定，則不允許更新。

## AdditionalIamPolicies(選擇性)

AdditionalIamPolicies:

- `Policy`: *string*

IAM 政策 Amazon 資源名稱 (ARN)。

為 Amazon Amazon EC2 指定 IAM 政策的亞馬遜資源名稱 (ARN) 清單。除了需要的權限之外，此清單還會附加至用於登入節點的根角色 AWS ParallelCluster。

IAM 政策名稱及其 ARN 不同。不能使用名稱。

如果已指定，則無法指定 InstanceRole 和 InstanceProfile 和設定。我們建議您使用 `AdditionalIamPolicies` 為已新增至 AWS ParallelCluster 需要的權限，且 InstanceRole 必須包含所有必要的權限。新增功能時，所需的權限通常會隨著發行版本而變更。

沒有預設值。

更新原則：如果變更此設定，則不允許更新。

`Policy`(必填[String])

更新原則：如果變更此設定，則不允許更新。

## Monitoring 區段

(選擇性) 指定叢集的監督設定。

Monitoring:

Logs:

CloudWatch:

Enabled: *boolean*

RetentionInDays: *integer*

DeletionPolicy: *string*

Rotation:

Enabled: *boolean*

Dashboards:

CloudWatch:

Enabled: *boolean*

DetailedMonitoring: *boolean*

Alarms:

Enabled: *boolean*

更新原則：更新期間不會分析此設定。

## Monitoring 屬性

### Logs(選擇性)

叢集的記錄設定。

更新原則：如果變更此設定，則不允許更新。

### CloudWatch(選擇性)

叢集的 CloudWatch 記錄檔設定。

更新原則：如果變更此設定，則不允許更新。

Enabled ( 必填 , Boolean )

如果 true，叢集記錄會串流至 CloudWatch 記錄檔。預設值為 true。

更新原則：如果變更此設定，則不允許更新。

RetentionInDays ( 可選 , Integer )

在記錄檔中保留 CloudWatch 錄事件的天數。預設值為 180。支援的值為 0、1、3、5、7、14、30、60、120、150、180、365、400、545、731、1827 和 3653。值 0 將使用預設的 CloudWatch 記錄保留設定，也就是永不過期。

更新原則：您可以在更新期間變更此設定。

DeletionPolicy ( 可選 , String )

指出刪除叢集時是否要刪除 CloudWatch 記錄檔上的記錄事件。可能的值為 Delete 和 Retain。預設值為 Retain。

更新原則：您可以在更新期間變更此設定。

### Rotation(選擇性)

叢集的記錄輪替設定。

更新原則：如果變更此設定，則不允許更新。

Enabled ( 必填 , Boolean )

如果 true，則會啟用記錄輪替。預設值為 true。當 AWS ParallelCluster 設定的記錄檔達到特定大小時，會旋轉該檔案並維護單一備份。如需詳細資訊，請參閱 [AWS ParallelCluster 配置的日誌輪替](#)。

更新原則：如果變更此設定，則不允許更新。

## Dashboards(選擇性)

叢集的儀表板設定。

更新原則：您可以在更新期間變更此設定。

## CloudWatch(選擇性)

叢集的 CloudWatch 儀表板設定。

更新原則：您可以在更新期間變更此設定。

Enabled ( 必填 , Boolean )

如果true，則會啟用 CloudWatch 儀表板。預設值為 true。

更新原則：您可以在更新期間變更此設定。

## DetailedMonitoring ( 可選 , Boolean )

如果設為true，則會為運算叢集 Amazon EC2 執行個體啟用詳細監控。啟用後，Amazon EC2 主控台會顯示每隔 1 分鐘監控執行個體的圖形。啟用此功能時會產生額外費用。預設值為 false。

如需詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的[啟用或關閉執行個體的詳細監控](#)。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### Note

DetailedMonitoring從 AWS ParallelCluster 版本 3.6.0 開始添加。

## Alarms(選擇性)

CloudWatch 叢集的警示。

更新原則：您可以在更新期間變更此設定。

Enabled(選擇性)

如果true，則會建立叢集的 CloudWatch 警示。預設值為 true。

更新原則：您可以在更新期間變更此設定。

**Note**

從 AWS ParallelCluster 版本 3.8.0 開始，系統會針對「頭節點」建立下列警示：  
Amazon EC2 運作 Health 檢查、CPU /記憶體/磁碟使用率，以及包括所有其他警示的複合警示。

## Tags 區段

(可選)，Array 定義由所有叢集資源使用 AWS CloudFormation 並傳播到所有叢集資源的標籤。若要取得更多資訊，請參閱《AWS CloudFormation 使用指南》中的[AWS CloudFormation 資源標籤](#)。

Tags:

- Key: *string*
- Value: *string*

更新原則：如果變更此設定，則不允許更新。

## Tags 屬性

Key ( 必填, String )

定義標籤的名稱。

更新原則：如果變更此設定，則不允許更新。

Value ( 必填, String )

定義標籤的值。

更新原則：如果變更此設定，則不允許更新。

## AdditionalPackages 區段

(選擇性) 用於識別要安裝的其他套件。

AdditionalPackages:

IntelSoftware:

IntelHpcPlatform: *boolean*



更新原則：如果變更此設定，則不允許更新。

## IntelSoftware

(選擇性) 定義 Intel 精選解決方案的組態。

```
IntelSoftware:  
  IntelHpcPlatform: boolean
```

更新原則：如果變更此設定，則不允許更新。

## IntelSoftware 屬性

IntelHpcPlatform ( 可選 , Boolean )

如果 true，表示接受 Intel 平行工作室的 [一般使用者授權合約](#)。這會導致英特爾並行工作室被安裝在頭節點上，並與計算節點共享。這增加了幾分鐘的時間，它需要頭節點引導。僅在 CentOS 7 上支援此 IntelHpcPlatform 設定。

更新原則：如果變更此設定，則不允許更新。

## DirectoryService 區段

### Note

在 3.1.1 AWS ParallelCluster 版本中添加了 Support。DirectoryService

(選擇性) 支援多重使用者存取之叢集的目錄服務設定。

AWS ParallelCluster 透過 [系統安全性服務常駐程式 \(SSSD\)](#) 支援的 [輕量型目錄存取通訊協定 \(LDAP\)](#) 來管理支援使用 [Active Directory \(AD\)](#) 的多個使用者存取叢集的權限。如需詳細資訊，請參閱《AWS Directory Service 使用者指南》中的 [什麼是 AWS Directory Service ?](#)。

建議您透過 TLS/SSL 使用 LDAP (簡稱為 LDAPS 的縮寫)，以確保任何可能的敏感資訊透過加密通道傳輸。

```
DirectoryService:  
  DomainName: string  
  DomainAddr: string
```

```

PasswordSecretArn: string
DomainReadOnlyUser: string
LdapTlsCaCert: string
LdapTlsReqCert: string
LdapAccessFilter: string
GenerateSshKeysForUsers: boolean
AdditionalSssdConfigs: dict

```

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

## DirectoryService 屬性

### Note

如果您打算 AWS ParallelCluster 在沒有網際網路存取權的單一子網路中使用，請參閱[AWS ParallelCluster 在沒有互聯網訪問的單個子網中](#)以取得其他需求。

### DomainName ( 必填 , String )

您用於身分識別資訊的作用中目錄 (AD) 網域。

DomainName 同時接受完整網域名稱 (FQDN) 和 LDAP 辨別名稱 (DN) 格式。

- FQDN 範例：corp.*example*.com
- LDAP 網域名稱範例：DC=*corp*,DC=*example*,DC=*com*

此屬性會對應至所呼叫的 sssd-ldap 參數。ldap\_search\_base

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

### DomainAddr ( 必填 , String )

指向做為 LDAP 伺服器之 AD 網域控制站的 URI 或 URI。URI 會對應至所呼叫的 SSSD-LDAP 參數。ldap\_uri 該值可以是 URI 的逗號分隔字串。若要使用 LDAP，您必須新增 ldap:// 至每個 URI 的開頭。

範例值：

```

ldap://192.0.2.0,ldap://203.0.113.0      # LDAP
ldaps://192.0.2.0,ldaps://203.0.113.0  # LDAPS without support for certificate
verification

```

```
ldaps://abcdef01234567890.corp.example.com # LDAPS with support for certificate
verification
192.0.2.0,203.0.113.0 # AWS ParallelCluster uses LDAPS by
default
```

如果您將 LDAPS 與憑證驗證搭配使用，則 URI 必須是主機名稱。

如果您在沒有憑證驗證或 LDAP 的情況下使用 LDAPS，則 URI 可以是主機名稱或 IP 位址。

使用透過 TLS/SSL (LDAPS) 進行的 LDAP，以避免透過未加密的通道傳輸密碼和其他敏感資訊。如果找 AWS ParallelCluster 不到協議，它將添加ldaps://到每個 URI 或主機名的開頭。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

PasswordSecretArn ( 必填, String )

包含DomainReadOnlyUser純文本密碼的 AWS Secrets Manager 密碼的 Amazon 資源名稱 ( ARN )。密碼的內容對應於所謂的 SSSD-LDAP 參數。ldap\_default\_authtok

#### Note

使用 AWS Secrets Manager 控制台創建密碼時，請務必選擇「其他類型的密碼」，選擇純文本，並且僅在密碼中包含密碼文本。

如需如何使用建立密碼 AWS Secrets Manager 的詳細資訊，請參閱[建立 AWS Secrets Manager 密碼](#)

LDAP 用戶端會在要求身分識別資訊DomainReadOnlyUser時，使用密碼來驗證 AD 網域。

如果使用者具有的權限 [DescribeSecret](#)，PasswordSecretArn則會驗證。

PasswordSecretArn如果指定的密碼存在，則為有效。如果使用者 IAM 政策不包  
含DescribeSecret，則PasswordSecretArn不會驗證並顯示警告訊息。如需詳細資訊，請參  
閱 [基本 AWS ParallelCluster pcluster使用者原則](#)。

當密碼的值變更時，叢集不會自動更新。若要更新新密碼值的叢集，您必須使用[the section called "pcluster update-compute-fleet"](#)命令停止計算叢集，然後從頭節點內執行下列命令。

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

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

## DomainReadOnlyUser ( 必填 , String )

驗證叢集使用者登入時，用來查詢 AD 網域以取得身分識別資訊的識別碼。它對應於所謂的 SSSD-LDAP 參數。ldap\_default\_bind\_dn針對此值使用您的 AD 身分識別資訊。

在節點上的特定 LDAP 用戶端所需的表單中指定識別碼：

- 微軟體:

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

- 簡單鉛：

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

[更新原則：必須停止運算叢集，才能變更此設定以進行更新。](#)

## LdapTlsCaCert ( 可選 , String )

憑證服務包的絕對路徑，其中包含憑證鏈結中為網域控制站發行憑證之每個憑證授權單位的憑證。它對應於所呼叫的 SSSD-LDAP 參數。ldap\_tls\_cacert

證書包是由 PEM 格式的不同證書串聯組成的文件，在 Windows 中也被稱為 DER Base64 格式。它是用來驗證做為 LDAP 伺服器之 AD 網域控制站的識別碼。

AWS ParallelCluster 不負責將憑證的初始放置到節點上。身為叢集管理員，您可以在建立叢集之後，在頭節點中手動設定憑證，也可以使用啟動[程序指令碼](#)。或者，您可以使用 Amazon 機器映像 (AMI)，其中包含在頭節點上設定的憑證。

S@@@ [imple AD](#) 不提供 LDAPS 支援。若要瞭解如何將 Simple AD 目錄與整合 AWS ParallelCluster，請參閱AWS 安全性部落格中的[如何為 Simple AD 設定 LDAPS 端點](#)。

[更新原則：必須停止運算叢集，才能變更此設定以進行更新。](#)

## LdapTlsReqCert ( 可選 , String )

指定要對 TLS 工作階段中的伺服器憑證執行哪些檢查。它對應於所謂的 SSSD-LDAP 參數。ldap\_tls\_reqcert

有效值：never、allow、try、demand 和 hard。

never、allow，並try啟用連線，即使發現憑證有問題，仍可繼續執行。

demand並在找不到憑證問題時hard啟用通訊以繼續。

如果叢集管理員使用的值不需要憑證驗證成功，系統管理員會傳回警告訊息。基於安全理由，我們建議您不要停用憑證驗證。

預設值為 `hard`。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

`LdapAccessFilter` ( 可選 , `String` )

指定篩選器，以限制對使用者子集的目錄存取。此內容會對應至所呼叫的 SSSD-LDAP 參數。 `ldap_access_filter` 您可以使用它將查詢限制為支援大量使用者的 AD。

此篩選器可以封鎖使用者對叢集的存取。但是，它不會影響被阻止用戶的可發現性。

如果設定此屬性，SSSD 參數 `access_provider` 會由 `ldap` 內部設定，AWS ParallelCluster 且不得由 [DirectoryService/AdditionalSssdConfigs](#) 設定修改。

如果省略此屬性，而且未在 [DirectoryService](#) 中指定自訂使用者存取權限 [AdditionalSssdConfigs](#)，則目錄中的所有使用者都可以存取叢集。

範例：

```

"!(cn=SomeUser*)" # denies access to every user with alias starting with "SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias starting with "SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".

```

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

`GenerateSshKeysForUsers` ( 可選 , `Boolean` )

定義叢集使用者在頭節點上的初始驗證之後，是否立即為叢集使用者 AWS ParallelCluster 產生 SSH 金鑰。

如果設定為 `true`，則會為每個使用者在頭節點上進行首次驗證後產生並儲存 SSH 金鑰 (如果不存在)。 `USER_HOME_DIRECTORY/.ssh/id_rsa`

對於尚未在 head 節點上進行身份驗證的用戶，在以下情況下可能會發生第一次身份驗證：

- 用戶首次使用自己或自己的密碼登錄到頭節點。
- 在頭節點中，`sudoer` 首次切換到用戶：`su USERNAME`
- 在頭節點中，`sudoer` 首次以用戶身份運行命令：`su -u USERNAME COMMAND`

使用者可以在後續登入叢集頭節點和計算節點時使用 SSH 金鑰。使用時 AWS ParallelCluster，叢集運算節點的密碼登入會停用設計。如果使用者尚未登入頭節點，則不會產生 SSH 金鑰，且使用者將無法登入計算節點。

預設值為 true。

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

AdditionalSssdConfigs ( 可選 , Dict )

鍵值對的字典，其中包含 SSSD 參數和值，以寫入叢集執行個體上的 SSSD 配置檔案。如需 SSSD 組態檔的完整說明，請參閱執行個體線上手冊頁面以 SSSD 及相關的設定檔。

SSSD 參數和值必須與下列清單中所述 AWS ParallelCluster 的 SSSD 組態相容。

- id\_provider 由 ldap 內部設定，AWS ParallelCluster 且不得修改。
- access\_provider 在指定 [DirectoryService](#) / AWS ParallelCluster 時設定 [LdapAccessFilter](#) 為 ldap 內部，且不得修改此設定。

如果省 [LdapAccessFilter](#) 略 [DirectoryService](#) /，它的 access\_provider 規格也被省略。例如，如果您設 access\_provider 定為 simple in [AdditionalSssdConfigs](#)，則不能指 [LdapAccessFilter](#) 定 [DirectoryService](#) /。

下列組態程式碼片段是的有效組態範例 AdditionalSssdConfigs。

此範例會啟用 SSSD 記錄檔的偵錯層級、將搜尋基礎限制為特定組織單位，以及停用認證快取。

```
DirectoryService:
...
AdditionalSssdConfigs:
  debug_level: "0xFFF0"
  ldap_search_base: OU=Users,OU=CORP,DC=corp,DC=example,DC=com
  cache_credentials: False
```

此範例指定 SSSD [simple](#) access\_provider 的組態。提供來自的 EngineeringTeam 使用者對目錄的存取權。 [DirectoryService](#) 在這種情況下，不得設置 [LdapAccessFilter](#)。

```
DirectoryService:
...
AdditionalSssdConfigs:
  access_provider: simple
  simple_allow_groups: EngineeringTeam
```

更新原則：必須停止運算叢集，才能變更此設定以進行更新。

## DeploymentSettings 區段

### Note

DeploymentSettings 已從 AWS ParallelCluster 版本 3.4.0 開始新增。

(選擇性) 指定部署設定規劃。

```
DeploymentSettings:  
  LambdaFunctionsVpcConfig:  
    SecurityGroupIds  
      - string  
    SubnetIds  
      - string  
  DisableSudoAccessForDefaultUser: Boolean  
  DefaultUserHome: string # 'Shared' or 'Local'
```

## DeploymentSettings 屬性

### LambdaFunctionsVpcConfig

(選擇性) 指定 AWS Lambda 功能 VPC 組態。如需詳細資訊，請參閱 [AWS Lambda 中的 VPC 組態](#) [AWS ParallelCluster](#)。

```
LambdaFunctionsVpcConfig:  
  SecurityGroupIds  
    - string  
  SubnetIds  
    - string
```

### LambdaFunctionsVpcConfig properties

SecurityGroupIds ( 必填, [String] )

附加至 Lambda 函數的 Amazon VPC 安全群組識別碼清單。

更新原則：如果變更此設定，則不允許更新。

## SubnetIds ( 必填 , [String] )

附加至 Lambda 函數的子網路識別碼清單。

更新原則：如果變更此設定，則不允許更新。

### Note

子網路和安全群組必須位於相同的 VPC 中。

## DisableSudoAccessForDefaultUser 財產

### Note

此組態選項僅支援Slurm叢集。

(選擇性) 如果True，則會停用預設使用者的 sudo 權限。這適用於叢集中的所有節點。

```
# Main DeploymentSettings section in config yaml(applyes to HN, CF and LN)
DeploymentSettings:
  DisableSudoAccessForDefaultUser: True
```

若要更新的值DisableSudoAccessForDefaultUser，您必須停止計算叢集和所有登入節點。

更新原則：必須停止計算叢集和登入節點，才能變更此設定以進行更新。

## DefaultUserHome 屬性

設定為時Shared，叢集將使用預設設定，並透過在叢集中共用預設使用者的目錄/home/<default user>。

設定為時Local，頭節點、登入節點和計算節點都會儲存個別的本機預設使用者目錄local/home/<default user>。

## 構建映像配置文件

AWS ParallelCluster 版本 3 使用 YAML 1.1 文件構建映像配置參數。請確認縮排是否正確，以減少組態錯誤。如需詳細資訊，請參閱 YAML 1.1 規格，網址為。<https://yaml.org/spec/1.1/>



這些組態檔是用來定義如何使用 EC2 Image Builder 建置您的自訂 AWS ParallelCluster AMI。使用 `pcluster build-image` 命令觸發自訂 AMI 建置程序。如需組態檔案範例，請參閱 [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test\\_imagebuilder\\_schema/test\\_imagebuilder\\_schema](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema)。

## 主題

- [構建映像配置文件屬性](#)
- [Build 區段](#)
- [Image 區段](#)
- [DeploymentSettings 區段](#)

## 構建映像配置文件屬性

Region ( 可選 , String )

指定 AWS 區域 作build-image業的。例如 us-east-2。

## Build 區段

(必要) 指定要在其中建立影像的組態。

```
Build:
  Imds:
    ImdsSupport: string
    InstanceType: string
    SubnetId: string
    ParentImage: string
  Iam:
    InstanceRole: string
    InstanceProfile: string
    CleanupLambdaRole: string
    AdditionalIamPolicies:
      - Policy: string
    PermissionsBoundary: string
  Components:
    - Type: string
      Value: string
  Tags:
    - Key: string
```

```
Value: string
SecurityGroupIds:
  - string
UpdateOsPackages:
  Enabled: boolean
```

## Build 屬性

InstanceType ( 必填 , String )

指定用來建置映像檔之執行個體的執行個體類型。

SubnetId ( 可選 , String )

指定要佈建執行個體以建置映像檔的現有子網路識別碼。提供的子網路需要網際網路存取。

### Warning

`pcluster build-image` 使用預設的 VPC。如果已刪除預設 VPC (可能是使用 AWS Control Tower 或 AWS 登陸區域), 則必須指定子網路 ID。

ParentImage ( 必填 , String )

指定基本影像。父映像可以是非 AWS ParallelCluster AMI, 也可以是相同版本的官方 AWS ParallelCluster AMI。您無法使用不同版本的 AWS ParallelCluster 官方或自訂 AMI AWS ParallelCluster。格式必須是圖像的 ARN `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion` 或 AMI ID `ami-12345678`。

SecurityGroupIds ( 可選 , [String] )

指定映像的安全性群組 ID 清單。

## Imsds

### Imsds 屬性

(選擇性) 指定 Amazon EC2 ImageBuilder 建置和測試執行個體中繼資料服務 (IMDS) 設定。

```
Imsds:
```

`ImdsSupport`: *string*

## ImdsSupport ( 可選 , String )

指定 Amazon EC2 ImageBuilder 建置和測試執行個體支援哪些 IMDS 版本。支援的值為 v2.0 和 v1.0。預設值為 v2.0。

如果設定 `ImdsSupport` 為 v1.0，則同時支援 IMDSv1 和 IMDSv2。

如果設定 `ImdsSupport` 為 v2.0，則僅支援 IMDSv2。

如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的使用 [IMDSv2](#)。

更新原則：如果變更此設定，則不允許更新。

### Note

從 AWS ParallelCluster 版本 3.7.0 開始，`ImdsSupport` 預設值為 v2.0。建議您在自訂動作呼叫中設定 `ImdsSupport` 為 IMDSv1，v2.0 並將其取代為 IMDSv2。Support [Imds/ImdsSupport](#) 添加了 3.3.0 AWS ParallelCluster 版本。

## Iam

### Iam 屬性

(選擇性) 指定映像組建的 IAM 資源。

`Iam`:

`InstanceRole`: *string*

`InstanceProfile`: *string*

`CleanupLambdaRole`: *string*

`AdditionalIamPolicies`:

- `Policy`: *string*

`PermissionsBoundary`: *string*

## InstanceProfile ( 可選 , String )

指定執行個體設定檔，以覆寫 EC2 Image Builder 執行個體的預設執行個體設定檔。

`InstanceProfile` 並 `InstanceRole` 且 `AdditionalIamPolicies` 不能一起指定。格式是 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`。

## InstanceRole ( 可選 , String )

指定執行個體角色，以覆寫 EC2 Image Builder 執行個體的預設執行個體角色。

InstanceProfile 並 InstanceRole 且 AdditionalIamPolicies 不能一起指定。格式是 `arn:Partition:iam::Account:role/RoleName`。

## CleanupLambdaRole ( 可選 , String )

IAM 角色的 ARN 用於支援 AWS CloudFormation 自訂資源的函數，該 AWS Lambda 函數會在建置完成時移除組建成品。Lambda 需要設定為允許擔任該角色的主體。格式是 `arn:Partition:iam::Account:role/RoleName`。

## AdditionalIamPolicies(選擇性)

指定要附加至用於產生自訂 AMI 的 EC2 Image Builder 執行個體的其他 IAM 政策。

```
AdditionalIamPolicies:  
- Policy: string
```

## Policy ( 可選 , [String] )

IAM 政策清單。格式是 `arn:Partition:iam::Account:policy/PolicyName`。

## PermissionsBoundary ( 可選 , String )

IAM 政策的 ARN，用作由 AWS ParallelCluster 建立的所有角色的許可界限。如需 IAM 許可界限的詳細資訊，請參閱 [IAM 使用者指南中 IAM 實體的許可界限](#)。格式是 `arn:Partition:iam::Account:policy/PolicyName`。

## Components

### Components 屬性

(選擇性) 除了預設提供的 ImageBuilder 元件外，還指定在 AMI 建置程序期間要使用的 Amazon EC2 元件 AWS ParallelCluster。這些組件可用於自定義 AMI 構建過程。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 定制](#)。

```
Components:  
- Type: string  
  Value: string
```

## Type ( 可選 , String )

指定元件的類型-值配對的類型。類型可以是arn或script。

## Value ( 可選 , String )

指定元件的類型-值配對的值。當類型為時arn，這是 EC2 Image Builder 元件的 ARN。當 type 為時script，這是指向建立 EC2 Image Builder 元件時要使用的指令碼的 https 或 s3 連結。

## Tags

### Tags 屬性

(選擇性) 指定要在用於建置 AMI 的資源中設定的標籤清單。

#### Tags:

- Key: *string*
- Value: *string*

## Key ( 可選 , String )

定義標籤的名稱。

## Value ( 可選 , String )

定義標籤的值。

## UpdateOsPackages

### UpdateOsPackages 屬性

(選擇性) 指定在安裝 AWS ParallelCluster 軟體堆疊之前是否更新作業系統。

#### UpdateOsPackages:

- Enabled: *boolean*

## Enabled ( 可選 , Boolean )

如果true，則在安裝 AWS ParallelCluster 軟件之前更新並重新啟動操作系統。預設值為 false。

**Note**

啟用時UpdateOsPackages，所有可用的作業系統套件都會更新，包括核心在內。身為客戶，您有責任確認更新是否與更新中未包含的 AMI 相依性相容。

例如，假設您正在為核心版本 Y.0 和某些元件 AWS ParallelCluster 版本 Z.0 隨附的 X.0 版本建置 AMI。假設可用的更新包含更新的核心版本 Y.1，而不更新元件 Z.0。在啟用之前UpdateOsPackages，您有責任確認元件 Z.0 是否支援 kernel Y.1。

## Image 區段

(選擇性) 定義映像版次的映像屬性。

**Image:**

**Name:** *string*

**RootVolume:**

**Size:** *integer*

**Encrypted:** *boolean*

**KmsKeyId:** *string*

**Tags:**

- **Key:** *string*

**Value:** *string*

## Image 屬性

Name ( 可選 , String )

指定 AMI 的名稱。如果未指定，則使用呼叫指 [pcluster build-image](#) 令時使用的名稱。

## Tags

### Tags 屬性

(選擇性) 指定影像的索引鍵值配對。

**Tags:**

- **Key:** *string*

**Value:** *string*

Key ( 可選 , String )

定義標籤的名稱。

Value ( 可選 , String )

定義標籤的值。

## RootVolume

### RootVolume 屬性

(選擇性) 指定映像的根磁碟區內容。

```
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  KmsKeyId: string
```

Size ( 可選 , Integer )

指定映像的根磁碟區大小 (以 GiB 為單位)。預設大小為[ParentImage](#)加上 27 GiB 的大小。

Encrypted ( 可選 , Boolean )

指定磁碟區是否已加密。預設值為 false。

KmsKeyId ( 可選 , String )

指定用來加密磁碟區之 AWS KMS 金鑰的 ARN。格式為  
"arn:*Partition*:kms:*Region*:*Account*:key/*KeyId*。

## DeploymentSettings 區段

(選擇性) 指定部署設定規劃。

```
DeploymentSettings:  
  LambdaFunctionsVpcConfig:  
    SecurityGroupIds  
      - string  
    SubnetIds  
      - string
```

## DeploymentSettings 屬性

### LambdaFunctionsVpcConfig

(選擇性) 指定 AWS Lambda 功能 VPC 組態。如需詳細資訊，請參閱 [AWS Lambda 中的 VPC 組態](#) [AWS ParallelCluster](#)。

#### LambdaFunctionsVpcConfig:

##### SecurityGroupIds

- *string*

##### SubnetIds

- *string*

### LambdaFunctionsVpcConfig properties

SecurityGroupIds ( 必填, [String] )

附加至 Lambda 函數的 Amazon VPC 安全群組識別碼清單。

更新原則：如果變更此設定，則不允許更新。

SubnetIds ( 必填, [String] )

附加至 Lambda 函數的子網路識別碼清單。

更新原則：如果變更此設定，則不允許更新。

#### Note

子網路和安全群組必須位於相同的 VPC 中。

#### Note

DeploymentSettings 已從 AWS ParallelCluster 版本 3.4.0 開始新增。

## AWS ParallelCluster API 參考資料

本節提供每個 AWS ParallelCluster API 動作的說明、語法和使用範例。



## 主題

- [建造影像](#)
- [createCluster](#)
- [刪除叢集](#)
- [deleteClusterInstances](#)
- [刪除影像](#)
- [描述叢集](#)
- [describeClusterInstances](#)
- [describeComputeFleet](#)
- [描述圖片](#)
- [getClusterLog](#)活動
- [getClusterStack](#)活動
- [getImageLog](#)活動
- [getImageStack](#)活動
- [列表集群](#)
- [listClusterLog](#)串流
- [listImageLog](#)串流
- [列表法師](#)
- [listOfficialImages](#)
- [更新叢集](#)
- [updateComputeFleet](#)

## 建造影像

在中建立自訂 AWS ParallelCluster 映像檔 AWS 區域。

## 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)

- [回應內文](#)
- [範例](#)

## 請求語法

```
POST /v3/images/custom
{
  "imageConfiguration": "string",
  "imageId": "string",
  "dryrun": boolean,
  "region": "string",
  "rollbackOnFailure": boolean,
  "supressValidators": [ "string" ],
  "validationFailureLevel": "string"
}
```

## 請求內文

### 映像配置

將映像組態設定為 YAML 文件。

類型：字串

必要：是

### imageId

要建置之映像檔的 ID。

類型：字串

必要：是

### 幹跑

如果設為 `true`，則僅執行請求驗證而不建立任何資源。使用此參數可驗證映像組態。預設值為 `false`。

類型：布林值

必要：否

## region

您運行命令以在其中構建映像的。AWS 區域

類型：字串

必要：否

## rollbackOnFailure

如果設定為true，則在映像檔建立失敗時發生映像堆疊復原。預設值為false。

類型：布林值

必要：否

## 抑制器驗證器

識別要抑制的一或多個組態驗證程式。

類型：字串清單

格式：(ALL|type:[A-Za-z0-9]+)

必要：否

## validationFailureLevel

造成映像檔建置失敗的最低驗證層級。預設值為ERROR。

類型：字串

有效值：INFO | WARNING | ERROR

必要：否

## 回應語法

```
{
  "image": {
    "imageId": "string",
    "ec2AmiInfo": {
      "amiId": "string"
    },
    "region": "string",
```

```
"version": "string",
"cloudformationStackArn": "string",
"imageBuildStatus": "BUILD_IN_PROGRESS",
"cloudformationStackStatus": "CREATE_IN_PROGRESS"
},
"validationMessages": [
  {
    "id": "string",
    "type": "string",
    "level": "INFO",
    "message": "string"
  }
]
}
```

## 回應內文

### image

#### imageId

影像的識別碼。

類型：字串

#### cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

#### cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmiInfo

### ami\_id

Amazon EC2 AMI ID。

類型：字串

### imageBuildStatus

映像建置狀態。

類型：字串

有效值：BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

### region

AWS 區域 該映像是內置的。

類型：字串

### version

用來建立映像檔的 AWS ParallelCluster 版本。

類型：字串

## 驗證訊息

驗證層級低於的訊息清單validationFailureLevel。訊息清單會在組態驗證期間收集。

### id

驗證程式識別碼。

類型：字串

### level

驗證層級。

類型：字串

有效值：INFO | WARNING | ERROR

## message

驗證訊息。

類型：字串

## type

驗證器的類型。

類型：字串

## 範例

### Python

請求

```
$ build_image(custom-image-id, custom-image-config.yaml)
```

二百個回應

```
{
  'image': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/custom-image-id/711b76b0-af81-11ec-a29f-0ee549109f1f',
    'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
    'image_build_status': 'BUILD_IN_PROGRESS',
    'image_id': 'custom-image-id',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

## createCluster

在中建立受管理叢集 AWS 區域。

主題

- [請求語法](#)
- [請求內文](#)

- [回應語法](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
POST /v3/clusters
{
  "clusterName": "string",
  "clusterConfiguration": "string",
  "dryrun": boolean,
  "region": "string",
  "rollbackOnFailure": boolean,
  "suppressValidators": [ "string" ],
  "validationFailureLevel": "string"
}
```

## 請求內文

### 叢集配置

作為 YAML 文件的叢集配置。

類型：字串

必要：是

### 叢集名稱

要建立的叢集名稱。

名稱必須以字母字元開頭。名稱最多可包含 60 個字元。如果啟用Slurm會計，名稱最多可包含 40 個字元。

類型：字串

必要：是

### 幹跑

如果設為true，則僅執行請求驗證而不建立任何資源。使用此參數可驗證叢集配置。預設值為false。

類型：布林值

必要：否

## region

叢集所在的。AWS 區域

類型：字串

必要：否

## rollbackOnFailure

如果設定為true，則會在叢集建立失敗時發生叢集堆疊復原。預設值為 true。

類型：布林值

必要：否

## 抑制器驗證器

識別要抑制的一或多個組態驗證程式。

類型：字串清單

格式：(ALL|type:[A-Za-z0-9]+)

必要：否

## validationFailureLevel

造成叢集建立失敗的最低驗證層級。預設值為 ERROR。

類型：字串

有效值：INFO | WARNING | ERROR

必要：否

## 回應語法

```
{  
  "cluster": {
```



```
"clusterName": "string",
"region": "string",
"version": "string",
"cloudformationStackArn": "string",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"clusterStatus": "CREATE_IN_PROGRESS",
"scheduler": {
  "type": "string",
  "metadata": {
    "name": "string",
    "version": "string"
  }
}
},
"validationMessages": [
  {
    "id": "string",
    "type": "string",
    "level": "INFO",
    "message": "string"
  }
]
}
```

## 回應內文

### 叢集名稱

叢集的名稱。

類型：字串

### cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

### cloudformationStackStatus

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE

| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## 叢集狀態

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## region

在中建立叢集的。AWS 區域

類型：字串

## 排程器

### 中繼資料

#### 排程器中繼資料

##### name

排程器的名稱。

類型：字串

##### version

排程器版本。

類型：字串

##### type

排程器類型。

類型：字串

## version

用來建立叢集的 AWS ParallelCluster 版本。

類型：字串

## 驗證訊息 ([])

驗證層級低於的訊息清單validationFailureLevel。訊息清單會在組態驗證期間收集。

id

驗證程式的識別碼。

類型：字串

level

類型：字串

有效值：INFO | WARNING | ERROR

message

驗證訊息。

類型：字串

type

驗證程式的類型。

類型：字串

## 範例

### Python

請求

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

二百個回應

```
{  
  'cluster': {  
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-  
east-1:123456789012:stack/cluster-3x/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
```

```
'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
'cluster_name': 'cluster-3x',
'cluster_status': 'CREATE_IN_PROGRESS',
'region': 'us-east-1',
'scheduler': {
  'type': 'slurm'
},
'version': '3.2.1'
}
}
```

## 刪除叢集

起始刪除叢集。

主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
DELETE /v3/clusters/{clusterName}
{
  "region": "string"
}
```

### 請求內文

#### 叢集名稱

叢集的名稱。

類型：字串

必要：是

## region

在中刪除叢集的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{
  "cluster": {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS",
    "clusterStatus": "DELETE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  }
}
```

## 回應內文

### 叢集

叢集執行個體清單

叢集名稱

叢集的名稱。

類型：字串

cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

cloudformationStackStatus

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

叢集狀態

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

region

在中建立叢集的。AWS 區域

類型：字串

排程器

中繼資料

排程器中繼資料。

name

排程器的名稱。

類型：字串

version

排程器版本

類型：字串

## type

排程器類型。

類型：字串

## version

用來建立叢集的 AWS ParallelCluster 版本。

類型：字串

## 範例

### Python

#### 請求

```
$ delete_cluster(cluster_name_3x)
```

#### 二百個回應

```
{
  'cluster': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
    'cloudformation_stack_status': 'DELETE_IN_PROGRESS',
    'cluster_name': 'cluster_name_3x',
    'cluster_status': 'DELETE_IN_PROGRESS',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

## deleteClusterInstances

啟動所有叢集運算節點的強制終止。此動作不支援 AWS Batch 叢集。

### 主題

- [請求語法](#)

- [請求內文](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
DELETE /v3/clusters/{clusterName}/instances
{
  "force": boolean,
  "region": "string"
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### force

如果設定為true，則在找不到具有指定名稱的叢集時強制刪除。預設值為false。

類型：布林值

必要：否

### region

叢集所在的。AWS 區域

類型：字串

必要：否

## 回應內文

無



## 範例

### Python

#### 請求

```
$ delete_cluster_instances(cluster_name_3x)
```

#### 二百個回應

#### 無

## 刪除影像

開始刪除自訂 AWS ParallelCluster 映像檔。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
DELETE /v3/images/custom/{imageId}  
{  
  "force": boolean,  
  "region": "string"  
}
```

### 請求內文

#### imageId

影像的識別碼。

類型：字串

必要：是

force

如果設定為true，則強制 AMI 刪除。如果有使用 AMI 的執行個體或 AMI 為共用，請使用此參數。  
預設值為 false。

類型：布林值

必要：否

region

圖像是在中創建的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{
  "image": {
    "imageId": "string",
    "ec2AmiInfo": {
      "amiId": "string"
    },
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "imageBuildStatus": "DELETE_IN_PROGRESS",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS"
  }
}
```

## 回應內文

image

cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

## cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmiInfo

AMIID

Amazon EC2 AMI ID。

類型：字串

## imageBuildStatus

映像建置狀態。

類型：字串

有效值：BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

## imageId

影像的識別碼。

類型：字串

## region

在中創建圖像。AWS 區域

類型：字串

## version

用來建立映像檔的 AWS ParallelCluster 版本。

類型：字串

## 範例

### Python

請求

```
$ delete_image(custom-image-id)
```

二百個回應

```
{
  'image': {
    'image_build_status': 'DELETE_IN_PROGRESS',
    'image_id': 'custom-image-id',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

## 描述叢集

取得現有叢集的詳細資訊。

主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/clusters/{clusterName}
{
  "region": "string"
```

```
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### region

叢集所在的。AWS 區域

類型：字串

必要：否

## 回應語法

### Note

failureReason已更改為failures從 AWS ParallelCluster 版本 3.5.0 開始。

```
{
  "clusterName": "string",
  "region": "string",
  "version": "string",
  "cloudFormationStackStatus": "CREATE_IN_PROGRESS",
  "clusterStatus": "CREATE_IN_PROGRESS",
  "scheduler": {
    "type": "string",
    "metadata": {
      "name": "string",
      "version": "string"
    }
  },
  "cloudformationStackArn": "string",
  "creationTime": "2019-08-24T14:15:22Z",
  "lastUpdatedTime": "2019-08-24T14:15:22Z",
```

```
"clusterConfiguration": {
  "url": "string"
},
"computeFleetStatus": "START_REQUESTED",
"tags": [
  {
    "key": "string",
    "value": "string"
  }
],
"headNode": {
  "instanceId": "string",
  "instanceType": "string",
  "launchTime": "2019-08-24T14:15:22Z",
  "privateIpAddress": "string",
  "publicIpAddress": "string",
  "state": "pending"
},
"failures": [
  {
    "failureCode": "string",
    "failureReason": "string"
  }
]
"loginNodes": {
  "status": "string",
  "address": "string",
  "scheme": "string",
  "healthyNodes": integer,
  "unhealthyNodes": integer
}
}
```

## 回應內文

### 叢集名稱

叢集的名稱。

類型：字串

### cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

## cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## 叢集配置

url

叢集配置檔案的 URL。

類型：字串

## 叢集狀態

叢集狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## computeFleetStatus

運算叢集狀態。

類型：字串

有效值：START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED  
| STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

## creationTime

建立叢集時的時間戳記。

類型：日期時間

lastUpdatedTime

上次更新叢集時的時間戳記。

類型：日期時間

region

在中建立叢集的。AWS 區域

類型：字串

標籤

與叢集相關聯的標籤清單。

金鑰

標籤名稱。

類型：字串

標籤

標籤值。

類型：字串

version

用來建立叢集的 AWS ParallelCluster 版本。

類型：字串

failures

叢集堆疊處於CREATE\_FAILED狀態時的失敗清單。

失敗代碼

叢集堆疊處於CREATE\_FAILED狀態時的失敗代碼。

類型：字串

失敗原因

叢集堆疊處於CREATE\_FAILED狀態時失敗的原因。



類型：字串

## 頭節點

叢集頭節點。

instanceId

亞馬遜 EC2 執行個體識別碼。

類型：字串

instanceType

Amazon EC2 實例類型。

類型：字串

launchTime

啟動亞馬遜 EC2 執行個體的時間。

類型：日期時間

privateIpAddress

叢集私有 IP 位址。

類型：字串

publicIpAddress

叢集公用 IP 位址。

類型：字串

state

頭節點實例狀態。

類型：字串

有效值：pending | running | shutting-down | terminated | stopping |  
stopped

## 排程器

中繼資料

排程器中繼資料。

**name**

排程器的名稱。

類型：字串

**version**

排程器版本。

類型：字串

**登入節點****status**

登錄節點的狀態。

類型：字串

有效值：PENDING | FAILED | ACTIVE

**address**

登入節點位址。

類型：字串

**結構描述**

登錄節點配置。

類型：字串

**結構描述**

健全狀況節點的數目。

類型：整數

**結構描述**

運作狀態不良的節點數目。

類型：整數

## type

排程器類型。

類型：字串

## 範例

### Python

請求

```
$ describe_cluster(cluster_name_3x)
```

二百個回應

```
{
  'cloud_formation_stack_status': 'CREATE_COMPLETE',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
  'cluster_configuration': {
    'url': 'https://parallelcluster-....'
  },
  'cluster_name': 'cluster_name_3x',
  'cluster_status': 'CREATE_COMPLETE',
  'compute_fleet_status': 'RUNNING',
  'creation_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
  'head_node': {
    'instance_id': 'i-abcdef01234567890',
    'instance_type': 't2.micro',
    'launch_time': datetime.datetime(2022, 3, 28, 22, 21, 56, tzinfo=tzlocal()),
    'private_ip_address': '172.31.56.3',
    'public_ip_address': '107.23.100.164',
    'state': 'running'
  },
  'last_updated_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
  'region': 'us-east-1',
  'tags': [
    {
      'key': 'parallelcluster:version', 'value': '3.2.1'
    }
  ]
}
```

```
    }  
  ],  
  'version': '3.2.1'  
}
```

## describeClusterInstances

說明屬於叢集的執行個體。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/clusters/{clusterName}/instances  
{  
  "nextToken": "string",  
  "nodeType": "string",  
  "queueName": "string",  
  "region": "string"  
}
```

### 請求內文

#### 叢集名稱

叢集的名稱。

類型：字串

必要：是

#### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

### 節點類型

依節點類型篩選執行個體。

類型：字串

有效值：HeadNode、ComputeNode、LoginNode

必要：否

### 佇列名稱

依佇列名稱篩選執行個體。

類型：字串

必要：否

### region

叢集所在的。AWS 區域

類型：字串

必要：否

### 回應語法

```
{
  "nextToken": "string",
  "instances": [
    {
      "instanceId": "string",
      "instanceType": "string",
      "launchTime": "2019-08-24T14:15:22Z",
      "privateIpAddress": "string",
      "publicIpAddress": "string",
      "state": "pending",
      "nodeType": "HeadNode",
      "queueName": "string"
    }
  ]
}
```

```
}  
]  
}
```

## 回應內文

### 執行個體

叢集執行個體的清單。

`instanceId`

亞馬遜 EC2 執行個體識別碼。

類型：字串

`instanceType`

Amazon EC2 實例類型。

類型：字串

`launchTime`

啟動亞馬遜 EC2 執行個體的時間。

類型：日期時間

`節點類型`

節點類型。

類型：字串

有效值：HeadNode、ComputeNode、LoginNode

`publicIpAddress`

叢集公用 IP 位址。

類型：字串

`佇列名稱`

Amazon EC2 執行個體支援節點的佇列名稱。

類型：字串

## state

節點亞馬遜 EC2 執行個體狀態。

類型：字串

有效值：pending | running | shutting-down | terminated | stopping | stopped

## nextToken

用於分頁請求的令牌。

類型：字串

## 範例

### Python

請求

```
$ describe_cluster_instances(cluster_name_3x)
```

二百個回應

```
{
  'instances': [
    {
      'instance_id': 'i-abcdef01234567890',
      'instance_type': 't2.micro',
      'launch_time': datetime.datetime(2022, 3, 30, 14, 2, 7, tzinfo=tzlocal()),
      'node_type': 'HeadNode',
      'private_ip_address': '192.0.2.5',
      'public_ip_address': '198.51.100.180',
      'state': 'running'
    }
  ]
}
```

## describeComputeFleet

描述運算叢集的狀態。

## 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
GET /v3/clusters/{clusterName}/computefleet
{
  "region": "string"
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### region

叢集所在的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{
  "status": "START_REQUESTED",
  "lastStatusUpdatedTime": "2019-08-24T14:15:22Z"
}
```



## 回應內文

### status

類型：字串

有效值：START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED  
| STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdated時間

表示上次狀態更新時間的時間戳記。

類型：日期時間

## 範例

### Python

請求

```
$ describe_compute_fleet(cluster_name_3x)
```

二百個回應

```
{  
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,  
    tzinfo=tzlocal()),  
  'status': 'RUNNING'  
}
```

## 描述圖片

取得有關現有映像檔的詳細資訊。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)

- [回應內文](#)
- [範例](#)

## 請求語法

```
GET /v3/images/custom/{imageId}  
{  
  "region": "string"  
}
```

## 請求內文

### imageId

影像的識別碼。

類型：字串

必要：是

### region

圖像是在中創建的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{  
  "imageId": "string",  
  "region": "string",  
  "version": "string",  
  "imageBuildStatus": "BUILD_IN_PROGRESS",  
  "imageBuildLogsArn": "string",  
  "cloudformationStackStatus": "CREATE_IN_PROGRESS",  
  "cloudformationStackStatusReason": "string",  
  "cloudformationStackArn": "string",  
  "creationTime": "2019-08-24T14:15:22Z",
```

```
"cloudformationStackCreationTime": "2019-08-24T14:15:22Z",
"cloudformationStackTags": [
  {
    "key": "string",
    "value": "string"
  }
],
"imageConfiguration": {
  "url": "string"
},
"imagebuilderImageStatus": "PENDING",
"imagebuilderImageStatusReason": "string",
"ec2AmiInfo": {
  "amiId": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "amiName": "string",
  "architecture": "string",
  "state": "PENDING",
  "description": "string"
}
}
```

## 回應內文

### imageId

要擷取詳細資訊的影像 ID。

類型：字串

### imageBuildStatus

映像建置狀態。

類型：字串

有效值：BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

## 映像配置

url

映像檔組態檔的 URL。

類型：字串

region

在中創建圖像。AWS 區域

類型：字串

version

用來建立映像檔的 AWS ParallelCluster 版本。

類型：字串

cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

cloudformationStackCreation時間

建立 CloudFormation 堆疊時的時間戳記。

類型：日期時間

cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## cloudformationStackStatus原因

CloudFormation 堆棧狀態的原因。

類型：字串

## cloudformationStackTags

CloudFormation 堆疊的標籤清單。

金鑰

標籤名稱。

類型：字串

value

標籤值。

類型：字串

## creationTime

建立影像時的時間戳記。

類型：日期時間

## ec2 AmiInfo

AMIID

Amazon EC2 AMI ID。

類型：字串

阿米娜美

Amazon EC2 AMI 名稱。

類型：字串

架構

Amazon EC2 AMI 架構。

類型：字串

## state

Amazon EC2 AMI 的狀態。

類型：字串

有效值：PENDING | AVAILABLE | INVALID | DEREGISTERED | TRANSIENT | FAILED | ERROR

## 標籤

Amazon EC2 AMI 標籤列表。

### 金鑰

標籤名稱。

類型：字串

### value

標籤值。

類型：字串

## imagebuilderImageStatus

狀 ImageBuilder 態。

類型：字串

有效值：PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING | INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

## imagebuilderImageStatus原因

ImageBuilder 圖像狀態的原因。

類型：字串

## imageBuildLogs阿恩

映像檔建置程序的日誌的 Amazon 資源名稱 (ARN)。

類型：字串

## 範例

### Python

#### 請求

```
$ describe_image(custom-image-id)
```

#### 二百個回應

```
{
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-image-id/6accc570-b080-11ec-845e-0e2dc6386985',
  'cloudformation_stack_creation_time': datetime.datetime(2022, 3, 30, 23, 23, 33,
731000, tzinfo=tzlocal()),
  'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
  'cloudformation_stack_tags': [
    {
      'key': 'parallelcluster:version', 'value': '3.2.1'
    },
    {
      'key': 'parallelcluster:image_name',
      'value': 'custom-image-id'
    },
    {
      'key': 'parallelcluster:custom-image-id',
      'value': 'custom-image-id'
    },
    {
      'key': 'parallelcluster:s3_bucket',
      'value': 'parallelcluster-abcdef01234567890-v1-do-not-delete'
    },
    {
      'key': 'parallelcluster:s3_image_dir',
      'value': 'parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0'
    },
    {
      'key': 'parallelcluster:build_log',
      'value': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-image-id'
    },
    {
      'key': 'parallelcluster:build_config',
```

```
    'value': 's3://parallelcluster-abcdef01234567890-v1-do-not-delete/parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-config.yaml'
  }
],
'image_build_logs_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/ParallelClusterImage-alinux2-image',
'image_build_status': 'BUILD_IN_PROGRESS',
'image_configuration': {
  'url': 'https://parallelcluster-abcdef01234567890-v1-do-not-delete.s3.amazonaws.com/parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-config.yaml?...'
},
'image_id': 'custom-image-id',
'imagebuilder_image_status': 'PENDING',
'region': 'us-east-1',
'version': '3.2.1'
}
```

## getClusterLog活動

擷取與記錄資料流相關聯的事件。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/clusters/{clusterName}/logstreams/{logStreamName}
{
  "endTime": datetime,
  "limit": float,
  "nextToken": "string",
  "region": "string",
  "startFromHead": boolean,
```



```
"startTime": datetime
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### logStreamName

日誌串流的名稱。

類型：字串

必要：是

### endTime

時間範圍的結束時間，以 ISO 8601 格式表示。不會包含時間戳記等於或晚於此時間的事件。

類型：日期時間

格式：2021-01-01T20:00:00Z

必要：否

### limit

傳回的記錄事件數目上限。如果未指定值，則最多可容納 1 MB 回應大小的記錄事件數目，最多可容納 10,000 個記錄事件。

類型：浮動

必要：否

### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

## region

叢集所在的。AWS 區域

類型：字串

必要：否

## startFromHead

如果設定為true，則會先傳回最早的記錄事件。如果值為false，則會先傳回最新的記錄事件。預設值為false。

類型：布林值

必要：否

## startTime

時間範圍的開始時間，以 ISO 8601 格式表示。包含時間戳記等於此時間或晚於此時間的事件。

類型：日期時間

格式：2021-01-01T20:00:00Z

必要：否

## 回應語法

```
{
  "nextToken": "string",
  "prevToken": "string",
  "events": [
    {
      "timestamp": "2019-08-24T14:15:22Z",
      "message": "string"
    }
  ]
}
```

## 回應內文

### 事件

已篩選事件的清單。

message

事件訊息。

類型：字串

timestamp

事件時間戳記。

類型：日期時間

nextToken

用於分頁請求的令牌。

類型：字串

上一个令牌

用於分頁請求的令牌。

類型：字串

## 範例

### Python

請求

```
$ get_cluster_log_events(cluster_name_3x, log_stream_name=ip-192-0-2-26.i-  
abcdef01234567890.cfn-init)
```

二百個回應

```
"events": [  
  {
```

```
    "message": "2022-09-22 16:40:15,127 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.us-east-1.amazonaws.com",
    "timestamp": "2022-09-22T16:40:15.127Z"
  },
  {
    "message": "2022-09-22 16:40:15,127 [DEBUG] Describing resource
HeadNodeLaunchTemplate in stack cluster_name_3x",
    "timestamp": "2022-09-22T16:40:15.127Z"
  },
  ...
]
```

## getClusterStack活動

擷取與叢集堆疊相關聯的事件。

### Note

從版本 3.6.0 開始，AWS ParallelCluster 使用巢狀堆疊來建立與佇列和計算資源相關聯的資源。GetClusterStackEventsAPI 和 `pcluster get-cluster-stack-events` 命令僅返回叢集主堆棧事件。您可以在 CloudFormation 主控台中檢視叢集堆疊事件，包括與佇列和計算資源相關的事件。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/clusters/{clusterName}/stackevents
{
  "nextToken": "string",
  "region": "string"
```

```
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

### region

叢集所在的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{  
  "nextToken": "string",  
  "events": [  
    {  
      "stackId": "string",  
      "eventId": "string",  
      "stackName": "string",  
      "logicalResourceId": "string",  
      "physicalResourceId": "string",  
      "resourceType": "string",  
      "timestamp": "2019-08-24T14:15:22Z",  
      "resourceStatus": "CREATE_IN_PROGRESS",  
      "resourceStatusReason": "string",
```

```
    "resourceProperties": "string",  
    "clientRequestToken": "string"  
  }  
]  
}
```

## 回應內文

### 事件

已篩選事件的清單。

`clientRequestToken`

傳遞至產生此事件之動作的權杖。

類型：字串

`eventId`

此事件的唯一識別碼。

類型：字串

`logicalResourceid`

範本中指定之資源的邏輯名稱。

類型：字串

`physicalResourceid`

與資源實體執行環境相關聯的名稱或唯一識別碼。

類型：字串

資源屬性

用來建立資源之屬性的 BLOB。

類型：字串

資源狀態

資源狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED  
| UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED  
| IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS |  
IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

resourceStatusReason

與資源相關聯的成功或失敗訊息。

類型：字串

resourceType

資源的類型。

類型：字串

堆疊 ID

堆疊執行個體的唯一 ID 名稱。

類型：字串

堆疊名

與堆疊相關聯的名稱。

類型：字串

timestamp

狀態更新的時間。

類型：日期時間

nextToken

用於分頁請求的令牌。

類型：字串

## 範例

Python

請求

```
$ get_cluster_stack_events(cluster_name_3x)
```

二百個回應

```
{
  'events': [
    {
      'event_id': '590b3820-b081-11ec-985e-0a7af5751497',
      'logical_resource_id': 'cluster_name_3x',
      'physical_resource_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
      'resource_status': 'CREATE_COMPLETE',
      'resource_type': 'AWS::CloudFormation::Stack',
      'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
      'stack_name': 'cluster_name_3x',
      'timestamp': datetime.datetime(2022, 3, 30, 23, 30, 13, 268000,
tzinfo=tzlocal())
    },
    ...
  ]
}
```

## getImageLog活動

擷取與映像組建相關聯的事件。

主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/images/custom/{imageId}/logstreams/{LogStreamName}
{
```



```
"endTime": datetime,  
"limit": float,  
"nextToken": "string",  
"region": "string",  
"startFromHead": boolean,  
"startTime": datetime  
}
```

## 請求內文

### imageId

影像的識別碼。

類型：字串

必要：是

### logStreamName

記錄資料流的名稱。

類型：字串

必要：是

### endTime

時間範圍的結束時間，以 ISO 8601 格式表示。不包含時間戳記等於或晚於此時間的事件。

類型：日期時間

格式：2021-01-01T20:00:00Z

必要：否

### limit

傳回的記錄事件數目上限。如果未指定值，則最多可容納 1 MB 回應大小的記錄事件數目，最多可容納 10,000 個記錄事件。

類型：浮動

必要：否

## nextToken

用於分頁請求的令牌。

類型：字串

必要：否

## region

AWS 區域 該圖像中。

類型：字串

必要：否

## startFromHead

如果設定為 `true`，請先傳回最早的記錄事件。如果設定為 `false`，請先傳回最新的記錄事件。預設值為 `false`。

類型：布林值

必要：否

## startTime

時間範圍的開始時間，以 ISO 8601 格式表示。包含時間戳記等於此時間或晚於此時間的事件。

類型：日期時間

格式：2021-01-01T20:00:00Z

必要：否

## 回應語法

```
{
  "nextToken": "string",
  "prevToken": "string",
  "events": [
    {
      "timestamp": "2019-08-24T14:15:22Z",
      "message": "string"
    }
  ]
}
```

```
}  
]  
}
```

## 回應內文

### 事件

已篩選事件的清單。

message

事件訊息。

類型：字串

timestamp

事件時間戳記。

類型：日期時間

nextToken

用於分頁請求的令牌。

類型：字串

上一个令牌

用於分頁請求的令牌。

類型：字串

## 範例

### Python

請求

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

二百個回應

```
"events": [  
  {  
    "message": "ExecuteBash: STARTED EXECUTION",  
    "timestamp": "2022-04-05T15:51:20.228Z"  
  },  
  {  
    "message": "ExecuteBash: Created temporary directory: /tmp/1234567890abcdef0",  
    "timestamp": "2022-04-05T15:51:20.228Z"  
  },  
  ...  
]
```

## getImageStack活動

擷取與映像檔組建之堆疊相關聯的事件。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/images/custom/{imageId}/stackevents  
{  
  "nextToken": "string",  
  "region": "string"  
}
```

### 請求內文

#### imageId

影像的識別碼。

類型：字串

必要：是

nextToken

用於分頁請求的令牌。

類型：字串

必要：否

region

AWS 區域 該圖像所在。

類型：字串

必要：否

## 回應語法

```
{
  "nextToken": "string",
  "events": [
    {
      "stackId": "string",
      "eventId": "string",
      "stackName": "string",
      "logicalResourceId": "string",
      "physicalResourceId": "string",
      "resourceType": "string",
      "timestamp": "2019-08-24T14:15:22Z",
      "resourceStatus": "CREATE_IN_PROGRESS",
      "resourceStatusReason": "string",
      "resourceProperties": "string",
      "clientRequestToken": "string"
    }
  ]
}
```

## 回應內文

### 事件

已篩選事件的清單。

## clientRequestToken

傳遞至產生此事件之動作的權杖。

類型：字串

## eventId

此事件的唯一識別碼。

類型：字串

## logicalResourceId

範本中指定之資源的邏輯名稱。

類型：字串

## physicalResourceId

與資源實體執行環境相關聯的名稱或唯一識別碼。

類型：字串

## 資源屬性

用來建立資源之屬性的 BLOB。

類型：字串

## 資源狀態

資源狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED  
| UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED  
| IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS |  
IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

## resourceStatusReason

與資源相關聯的成功或失敗訊息。

類型：字串

## resourceType

資源的類型。

類型：字串

## 堆疊 ID

堆疊執行個體的唯一 ID 名稱。

類型：字串

## 堆疊名

與堆疊相關聯的名稱。

類型：字串

## timestamp

狀態更新的時間。

類型：日期時間

## nextToken

用於分頁請求的令牌。

類型：字串

## 範例

### Python

請求

```
$ get_image_stack_events(image_id)
```

二百個回應

```
{
  'events': [
    {
      'event_id': 'ParallelClusterImage-
CREATE_IN_PROGRESS-2022-03-30T23:26:33.499Z',
      'logical_resource_id': 'ParallelClusterImage',
```

```
    'physical_resource_id': 'arn:aws:imagebuilder:us-east-1:123456789012:image/parallelclusterimage-alinux2-image/3.2.1/1',
    'resource_properties': {
      "InfrastructureConfigurationArn": "arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/parallelclusterimage-6accc570-b080-11ec-845e-0e2dc6386985",
      "ImageRecipeArn": "arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/parallelclusterimage-alinux2-image/3.2.1",
      "DistributionConfigurationArn": "arn:aws:imagebuilder:us-east-1:123456789012:distribution-configuration/parallelclusterimage-6accc570-b080-11ec-845e-0e2dc6386985",
      "EnhancedImageMetadataEnabled": "false",
      "Tags": {
        "parallelcluster:image_name": "alinux2-image", "parallelcluster:image_id": "alinux2-image"
      }
    },
    'resource_status': 'CREATE_IN_PROGRESS',
    'resource_status_reason': 'Resource creation Initiated',
    'resource_type': 'AWS::ImageBuilder::Image',
    'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/alinux2-image/6accc570-b080-11ec-845e-0e2dc6386985',
    'stack_name': 'alinux2-image',
    'timestamp': datetime.datetime(2022, 3, 30, 23, 26, 33, 499000, tzinfo=tzlocal())
  },
  ...
]
```

## 列表集群

擷取現有叢集的清單。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)



## 請求語法

```
GET /v3/clusters
{
  "clusterStatus": "string",
  "nextToken": "string",
  "region": "string"
}
```

## 請求內文

### 叢集狀態

依叢集狀態篩選。預設值為所有叢集。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_FAILED

必要：否

### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

### region

叢 AWS 區域 集的。

類型：字串

必要：否

## 回應語法

```
{
```

```
"nextToken": "string",
"clusters": [
  {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "clusterStatus": "CREATE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  }
]
}
```

## 回應內文

### 叢集

#### cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

#### cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## 叢集名稱

叢集的名稱。

類型：字串

## 叢集狀態

叢集狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## 排程器

### 中繼資料

排程器中繼資料。

name

排程器的名稱。

類型：字串

version

排程器版本。

類型：字串

type

排程器的類型。

類型：字串

region

在中建立叢集的。AWS 區域

類型：字串

version

用來建立叢集的 AWS ParallelCluster 版本。

類型：字串

nextToken

用於分頁請求的令牌。

類型：字串

## 範例

Python

請求

```
$ list_clusters()
```

二百個回應

```
{
  'clusters':
  [
    {
      'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
      'cloudformation_stack_status': 'CREATE_COMPLETE',
      'cluster_name': 'cluster_name_3x',
      'cluster_status': 'CREATE_COMPLETE',
      'region': 'us-east-1',
      'version': '3.2.1'
    },
    ...
  ]
}
```

## listClusterLog串流

擷取與叢集相關聯的記錄資料流清單。

主題

- [請求語法](#)

- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
GET /v3/clusters/{clusterName}/logstreams
{
  "filters": [ "string" ],
  "nextToken": "string",
  "region": "string"
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### 篩選條件

篩選記錄資料流。

接受的篩選器有：

- private-dns-name：執行個體的私有 DNS 名稱的簡短格式 (例如 IP -10-0-101)。
- 節點類型：有效值：。HeadNode

類型：唯一的字符串數組

格式：Name=a,Values=1 Name=b,Values=2,3

必要：否

### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

region

叢集所在的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{
  "nextToken": "string",
  "logStreams": [
    {
      "logStreamName": "string",
      "creationTime": "2019-08-24T14:15:22Z",
      "firstEventTimestamp": "2019-08-24T14:15:22Z",
      "lastEventTimestamp": "2019-08-24T14:15:22Z",
      "lastIngestionTime": "2019-08-24T14:15:22Z",
      "uploadSequenceToken": "string",
      "logStreamArn": "string"
    }
  ]
}
```

## 回應內文

### 日誌流

記錄資料流的清單。

creationTime

建立串流的時間。

類型：日期時間

firstEventTimestamp

串流第一個事件的時間。

類型：日期時間

lastEventTimestamp

串流最後一個事件的時間。lastEventTime 值會在最終一致性的基礎上進行更新。它通常會在擷取後不到一小時的時間內更新，但在極少數情況下可能需要更長的時間。

類型：日期時間

lastIngestionTime

上次擷取時間。

類型：日期時間

logStreamArn

日誌資料流的 Amazon 資源名稱 (ARN)。

類型：字串

logStreamName

記錄資料流的名稱。

類型：字串

uploadSequenceToken

序列令牌。

類型：字串

nextToken

用於分頁請求的令牌。

類型：字串

## 範例

Python

請求

```
$ list_cluster_log_streams(cluster_name_3x)
```

## 二百個回應

```
{
  'log_streams': [
    {
      'creation_time': datetime.datetime(2022, 3, 30, 14, 7, 34, 354000,
tzinfo=tzlocal()),
      'first_event_timestamp': datetime.datetime(2022, 3, 30, 14, 6, 41, 444000,
tzinfo=tzlocal()),
      'last_event_timestamp': datetime.datetime(2022, 3, 30, 14, 25, 55, 462000,
tzinfo=tzlocal()),
      'last_ingestion_time': datetime.datetime(2022, 3, 30, 14, 49, 50, 62000,
tzinfo=tzlocal()),
      'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
parallelcluster/cluster_name_3x:log-stream:ip-192-0-2-26.i-abcdef01234567890.cfn-
init',
      'log_stream_name': 'ip-192-0-2-26.i-abcdef01234567890.cfn-init',
      ...
      'upload_sequence_token': '####'
    },
    ...
  ]
}
```

## listImageLog串流

擷取與映像相關聯的記錄資料流清單。

### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

### 請求語法

```
GET /v3/images/custom/{imageId}/logstreams
```



```
{
  "nextToken": "string",
  "region": "string"
}
```

## 請求內文

### imageId

影像的識別碼。

類型：字串

必要：是

### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

### region

AWS 區域 該圖像所在。

類型：字串

必要：否

## 回應語法

```
{
  "nextToken": "string",
  "logStreams": [
    {
      "logStreamName": "string",
      "creationTime": "2019-08-24T14:15:22Z",
      "firstEventTimestamp": "2019-08-24T14:15:22Z",
      "lastEventTimestamp": "2019-08-24T14:15:22Z",
      "lastIngestionTime": "2019-08-24T14:15:22Z",
      "uploadSequenceToken": "string",
    }
  ]
}
```

```
    "logStreamArn": "string"  
  }  
]  
}
```

## 回應內文

### 日誌流

記錄資料流的清單。

#### creationTime

建立串流的時間。

類型：日期時間

#### firstEventTimestamp

串流中第一個事件的時間。

類型：日期時間

#### lastEventTimestamp

串流最後一個事件的時間。lastEventTime 值會在最終一致性的基礎上進行更新。它通常會在擷取後不到一小時的時間內更新，但在極少數情況下可能需要更長的時間。

類型：日期時間

#### lastIngestionTime

上次擷取時間。

類型：日期時間

#### logStreamArn

日誌資料流的 Amazon 資源名稱 (ARN)。

類型：字串

#### logStreamName

日誌串流的名稱。

類型：字串

uploadSequenceToken

序列令牌。

類型：字串

下一個令牌

用於分頁請求的令牌。

類型：字串

## 範例

### Python

請求

```
$ list_image_log_streams(custom-image-id)
```

二百個回應

```
{
  'log_streams': [
    {
      'creation_time': datetime.datetime(2022, 3, 29, 20, 29, 24, 875000,
tzinfo=tzlocal()),
      'first_event_timestamp': datetime.datetime(2022, 3, 29, 20, 29, 24, 775000,
tzinfo=tzlocal()),
      'last_event_timestamp': datetime.datetime(2022, 3, 29, 20, 38, 23, 944000,
tzinfo=tzlocal()),
      'last_ingestion_time': datetime.datetime(2022, 3, 29, 20, 51, 56, 26000,
tzinfo=tzlocal()),
      'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image:log-stream:3.2.1/1',
      'log_stream_name': '3.2.1/1',
      'upload_sequence_token': '####'
    },
    ...
  ]
}
```

# 列表法師

擷取現有自訂映像檔的清單。

## 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
GET /images/custom
{
  "imageStatus": "string",
  "nextToken": "string",
  "region": "string"
}
```

## 請求內文

### 影像狀態

依提供的狀態篩選影像。

類型：字串

有效值：AVAILABLE | PENDING | FAILED

必要：是

### nextToken

用於分頁請求的令牌。

類型：字串

必要：否

## region

AWS 區域 該圖像是在。

類型：字串

必要：否

## 回應語法

```
{
  "nextToken": "string",
  "images": [
    {
      "imageId": "string",
      "ec2AmiInfo": {
        "amiId": "string"
      },
      "region": "string",
      "version": "string",
      "cloudformationStackArn": "string",
      "imageBuildStatus": "BUILD_IN_PROGRESS",
      "cloudformationStackStatus": "CREATE_IN_PROGRESS"
    }
  ]
}
```

## 回應內文

### images

圖像列表。

#### cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

#### cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

#### ec2 AmiInfo

ami\_id

Amazon EC2 AMI ID。

類型：字串

#### imageBuildStatus

映像建置狀態。

有效值：BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

類型：字串

#### imageId

影像的識別碼。

類型：字串

#### region

在中創建圖像。AWS 區域

類型：字串

#### version

用來建立映像檔的 AWS ParallelCluster 版本。

類型：字串

#### nextToken

用於分頁請求的令牌。

類型：字串

## 範例

### Python

#### 請求

```
$ list_images("AVAILABLE")
```

#### 二百個回應

```
{
  'images': [
    {
      'ec2_ami_info': {
        'ami_id': 'ami-abcdef01234567890'
      },
      'image_build_status': 'BUILD_COMPLETE',
      'image_id': 'custom-image',
      'region': 'us-east-1',
      'version': '3.2.1'
    }
  ]
}
```

## listOfficialImages

檢索 AWS ParallelCluster 官方圖像列表。

#### 主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

#### 請求語法

```
GET /v3/images/official
```

```
{
  "architecture": "string",
  "os": "string",
  "region": "string"
}
```

## 請求內文

### 架構

依架構篩選。預設值為無篩選。

類型：字串

有效值：x86\_64 | arm64

必要：否

### OS

依作業系統分佈篩選。預設值為無篩選。

類型：字串

有效值：alinux2 | centos7 | ubuntu2204 | ubuntu2004 | rhel8

必要：否

### region

AWS 區域 該官方圖像中列出。

類型：字串

必要：否

## 回應語法

```
{
  "images": [
    {
      "architecture": "string",
      "amiId": "string",
      "name": "string",
    }
  ]
}
```



```
    "os": "string",
    "version": "string"
  }
]
```

## 回應內文

### images

#### AMIID

AMI 的 ID。

類型：字串

#### 架構

AMI 架構。

類型：字串

#### name

AMI 的名稱。

類型：字串

#### os

AMI 作業系統。

類型：字串

#### version

版 AWS ParallelCluster 本。

類型：字串

## 範例

### Python

#### 請求

```
$ list_official_images()
```

二百個回應

```
{
  'images': [
    {
      'ami_id': 'ami-015cfefb4e0d6306b2',
      'architecture': 'x86_64',
      'name': 'aws-parallelcluster-3.2.1-ubuntu-2004-lts-hvm-x86_64-202202261505 '
      '2022-02-26T15-08-34.759Z',
      'os': 'ubuntu2004',
      'version': '3.2.1'
    },
    ...
  ]
}
```

## 更新叢集

更新叢集。

主題

- [請求語法](#)
- [請求內文](#)
- [回應語法](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
PUT /v3/clusters/{clusterName}
{
  "clusterConfiguration": "string",
  "dryrun": boolean,
  "forceUpdate": boolean,
  "region": "string",
```

```
"suppressValidators": "string",  
"validationFailureLevel": "string"  
}
```

## 請求內文

### 叢集配置

作為 YAML 文件的叢集配置。

必要：是

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### 幹跑

如果設定為 `true`，則僅執行請求驗證而不建立任何資源。使用此參數可驗證叢集配置和更新需求。預設值為 `false`。

類型：布林值

必要：否

### 強制更新

如果設定為 `true`，則忽略更新驗證錯誤並強制更新。預設值為 `false`。

類型：布林值

必要：否

### region

叢集所在的。AWS 區域

類型：字串

必要：否

## 抑制器驗證器

識別要抑制的一或多個組態驗證程式。

類型：字串

格式：(ALL|type:[A-Za-z0-9]+)

必要：否

有效值範例：currentValue、requestedValue、message

### validationFailureLevel

造成更新失敗的最低驗證層級。

類型：字串

有效值：INFO | WARNING | ERROR

必要：否

## 回應語法

```
{
  "cluster": {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
    "clusterStatus": "UPDATE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  },
  "validationMessages": [
    {
      "id": "string",
```

```
    "type": "string",
    "level": "INFO",
    "message": "string"
  }
],
"changeSet": [
  {
    "parameter": "string",
    "currentValue": "string",
    "requestedValue": "string"
  }
]
}
```

## 回應內文

### changeSet

叢集更新所設定的變更。

#### 目前值

要更新的參數的當前值。

類型：字串

#### 參數

要更新的參數。

類型：字串

#### 請求的值

要更新的參數的請求值。

類型：字串

### 叢集

#### cloudformationStackArn

主 CloudFormation 堆棧的 Amazon 資源名稱 ( ARN ) 。

類型：字串

## cloudformationStackStatus

CloudFormation 堆疊狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## 叢集名稱

叢集的名稱。

類型：字串

## 叢集狀態

叢集狀態。

類型：字串

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## region

在中建立叢集的。AWS 區域

類型：字串

## 排程器

### 中繼資料

排程器中繼資料。

name

排程器的名稱。

類型：字串

version

排程器版本。

類型：字串

type

排程器類型。

類型：字串

version

AWS ParallelCluster 用來建立叢集的版本。

類型：字串

驗證訊息

驗證層級低於的訊息清單validationFailureLevel。訊息清單會在組態驗證期間收集。

id

驗證程式的識別碼。

類型：字串

level

驗證層級。

類型：字串

有效值：INFO | WARNING | ERROR

message

驗證訊息。

類型：字串

type

驗證器的類型。

類型：字串

## 範例

### Python

#### 請求

```
$ update_cluster(cluster_name_3x, path/config-file.yaml)
```

#### 二百個回應

```
{
  'change_set': [
    {
      'current_value': '10',
      'parameter':
      'Scheduling.SlurmQueues[queue1].ComputeResources[t2micro].MaxCount',
      'requested_value': '15'
    }
  ],
  'cluster': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/test-api-cluster/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
    'cloudformation_stack_status': 'UPDATE_IN_PROGRESS',
    'cluster_name': 'cluster-3x',
    'cluster_status': 'UPDATE_IN_PROGRESS',
    'region': 'us-east-1',
    'scheduler': {
      'type': 'slurm'
    },
    'version': '3.2.1'
  }
}
```

## updateComputeFleet

更新叢集運算叢集的狀態。

#### 主題

- [請求語法](#)
- [請求內文](#)



- [回應語法](#)
- [回應內文](#)
- [範例](#)

## 請求語法

```
PATCH /v3/clusters/{clusterName}/computefleet
{
  "status": "string",
  "region": "string"
}
```

## 請求內文

### 叢集名稱

叢集的名稱。

類型：字串

必要：是

### status

運算叢集狀態。

類型：字串

有效值：START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

必要：是

### region

叢集所在的。AWS 區域

類型：字串

必要：否

## 回應語法

```
{
```

```
"status": "START_REQUESTED",
"lastStatusUpdatedTime": "2019-08-24T14:15:22Z"
}
```

## 回應內文

### status

運算叢集狀態。

類型：字串

有效值：START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED  
| STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdated時間

代表上次狀態更新時間的時間戳記。

類型：日期時間

## 範例

### Python

請求

```
$ update_compute_fleet(cluster_name_3x, "START_REQUESTED")
```

二百個回應

```
{
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
    tzinfo=tzlocal()),
  'status': 'START_REQUESTED'
}
```

## AWS ParallelClusterPython 函式庫

開始於AWS ParallelCluster版本 3.5.0，您可以訪問AWS ParallelCluster與AWS ParallelCluster  
蟒蛇庫。您可以存取AWS ParallelCluster您的圖書館pccluster環境或從內AWS Lambda執行

階段。瞭解如何存取AWS ParallelClusterAPI 透過使用AWS ParallelCluster蟒蛇庫。該AWS ParallelClusterPython 程式庫提供的功能與AWS ParallelClusterAPI 可提供。

該AWS ParallelClusterPython 庫操作和參數在轉換為時鏡像 API 參數的那些snake\_case沒有大寫字母。

## 主題

- [AWS ParallelCluster程式庫授權](#)
- [安裝AWS ParallelCluster蟒蛇圖書館](#)
- [叢集 API 作業](#)
- [運算叢集 API 作業](#)
- [叢集和堆疊記錄作業](#)
- [影像 API 作業](#)
- [影像和堆疊記錄作業](#)
- [範例](#)
- [AWS Lambda為了AWS ParallelCluster蟒蛇圖書館](#)

## AWS ParallelCluster程式庫授權

使用任何對 boto3 有效的標準方式來指定認證。如需詳細資訊，請參閱[肉毒桿菌素 3 文檔](#)。

## 安裝AWS ParallelCluster蟒蛇圖書館

1. 安裝pcluster按照中給出的說明進行操作，CLI 版本 3.5.0 或更高版本[設定 AWS ParallelCluster](#)。
2. 匯入pcluster模組並開始使用程式庫，如下列範例所示：

```
import pcluster.lib as pc
pc.create_cluster(cluster_name="mycluster", cluster_configuration="config.yaml")
```

## 叢集 API 作業

### 主題

- [list\\_clusters](#)
- [create\\_cluster](#)

- [delete\\_cluster](#)
- [describe\\_cluster](#)
- [update\\_cluster](#)

## list\_clusters

```
list_clusters(region, next_token, cluster_status)
```

擷取現有叢集的清單。

參數：

### region

列出部署到指定的叢集AWS 區域。

### next\_token

用於分頁請求的令牌。

### cluster\_status

依叢集狀態篩選。預設值是列出所有叢集。

有效值：CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE |  
UPDATE\_FAILED

## create\_cluster

```
create_cluster(cluster_name, cluster_configuration, region, suppress_validators,  
validation_failure_level, dry_run, rollback_on_failure, wait)
```

在給定的區域中創建一個集群。

參數：

### cluster\_name (必要)

叢集名稱。

## **cluster\_configuration** (必要)

將群集配置作為 Python 數據類型。

### **region**

叢集AWS 區域。

### **suppress\_validators**

識別要隱藏的一或多個叢集配置驗證程式。

格式：(ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

造成叢集建立失敗的最低驗證層級。預設值為 ERROR。

有效值：INFO |WARNING | ERROR。

### **dry\_run**

在不建立任何資源的情況下執行要求驗證。您可以使用它來驗證叢集配置。預設值為 False。

### **rollback\_on\_failure**

如果設定為True,AWS ParallelCluster在失敗時自動啟動叢集堆疊復原。預設值為 True。

### **wait**

如果設定為True,AWS ParallelCluster等待作業完成。預設值為 False。

## **delete\_cluster**

```
delete_cluster(cluster_name, region, wait)
```

刪除指定區域中的叢集。

參數：

### **cluster\_name** (必要)

叢集名稱。

### **region**

叢集AWS 區域。

## wait

如果設定為 True，等待作業完成。預設值為 False。

## describe\_cluster

```
describe_cluster(cluster_name, region)
```

取得有關現有叢集的詳細資訊。

參數：

**cluster\_name** (必要)

叢集名稱。

**region**

叢集AWS 區域。

## update\_cluster

```
update_cluster(cluster_name, cluster_configuration, suppress_validators,  
validation_failure_level, region, force_update, dry_run, wait)
```

更新指定區域中的叢集。

參數：

**cluster\_name** (必要)

叢集名稱。

**cluster\_configuration** (必要)

將群集配置作為 Python 數據類型。

**suppress\_validators**

識別要隱藏的一或多個叢集配置驗證程式。

格式：(ALL | type:[A-Za-z0-9]+)

## validation\_failure\_level

造成叢集更新失敗的最低驗證層級。預設值為 ERROR。

有效值：INFO | WARNING | ERROR

## region

叢集AWS 區域。

## dry\_run

執行要求驗證，而不建立或更新任何資源。您可以使用它來驗證叢集配置。預設值為 False。

## force\_update

如果設定為True，透過忽略更新驗證錯誤來強制更新。預設值為 False。

## wait

如果設定為True，等待作業完成。預設值為 False。

## 運算叢集 API 作業

### 主題

- [describe\\_compute\\_fleet](#)
- [update\\_compute\\_fleet](#)
- [delete\\_cluster\\_instances](#)
- [describe\\_cluster\\_instances](#)

## describe\_compute\_fleet

```
describe_compute_fleet(cluster_name, region)
```

說明指定叢集之叢集運算叢集的狀態。

參數：

### cluster\_name (必要)

叢集名稱。

## region

描述部署至指定之叢集的計算叢集狀態AWS 區域。

## update\_compute\_fleet

```
update_compute_fleet(cluster_name, status, region)
```

更新叢集運算叢集的狀態。

參數：

### cluster\_name (必要)

叢集名稱。

### status (必要)

要更新為的狀態。

有效值：START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

### region

叢集AWS 區域。

## delete\_cluster\_instances

```
delete_cluster_instances(cluster_name, region, force)
```

刪除指定區域中的叢集。

參數：

### cluster\_name (必要)

叢集名稱。

### region

叢集AWS 區域。



## force

如果設定為True，當具有給定的集群時強制刪除cluster\_name沒有找到。預設值為 False。

## describe\_cluster\_instances

```
describe_cluster_instances(cluster_name, region, next_token, node_type, queue_name)
```

描述叢集的執行個體。

參數：

**cluster\_name** (必要)

叢集名稱。

**region**

叢集AWS 區域。

**next\_token**

用於分頁請求的令牌。

**node\_type**

篩選執行個體的依據node\_type。

有效值：HeadNode | ComputeNode

**queue\_name**

依佇列名稱篩選執行個體。

## 叢集和堆疊記錄作業

主題

- [list\\_cluster\\_log\\_streams](#)
- [get\\_cluster\\_log\\_events](#)
- [get\\_cluster\\_stack\\_events](#)

## list\_cluster\_log\_streams

```
list_cluster_log_streams(cluster_name, region, filters, next_token)
```

列出指定叢集的記錄資料流。

參數：

### cluster\_name (必要)

叢集名稱。

### region

叢集AWS 區域。

### filters

篩選叢集日誌串流。

格式：'Name=a,Values=1 Name=b,Values=2,3'

接受的篩選器：

code-dns-name

執行個體私有 DNS 名稱的簡短格式；例如，ip-10-0-0-101。

節點類型

節點類型。

有效值：HeadNode

### next\_token

用於分頁請求的令牌。

## get\_cluster\_log\_events

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,  
start_from_head, limit, start_time, end_time)
```

取得指定叢集和記錄資料流的記錄事件。

參數：

**cluster\_name** (必要)

叢集名稱。

**log\_stream\_name** (必要)

記錄資料流名稱。

**region**

叢集AWS 區域。

**next\_token**

用於分頁請求的令牌。

**start\_from\_head**

如果設定為True,AWS ParallelCluster首先傳回最早的記錄事件。如果設定為False , 它會先傳回最新的記錄事件。預設值為 False。

**limit**

傳回的記錄事件數目上限。如果未指定值 , 則最多可容納 1 MB 回應大小的記錄檔數目 , 最多 10,000 個記錄事件。

**start\_time**

記錄事件的時間範圍開始 , 以 ISO 8601 格式表示 ; 例如 , '2021-01-01T20:00:00Z'。包含時間戳記等於或晚於此時間的事件。

**end\_time**

記錄事件的時間範圍結束時間 , 以 ISO 8601 格式表示 ; 例如 , '2021-01-01T20:00:00Z'。不會包含時間戳記等於或晚於此時間的事件。

## get\_cluster\_stack\_events

```
get_cluster_stack_events(cluster_name, region, next_token)
```

獲取給定集群的堆棧事件。

參數：

**cluster\_name** (必要)

叢集名稱。

**region**

叢集AWS 區域。

**next\_token**

用於分頁請求的令牌。

## 影像 API 作業

主題

- [list\\_images](#)
- [build\\_image](#)
- [delete\\_image](#)
- [describe\\_image](#)

### list\_images

```
list_images(image_status, region, next_token)
```

檢索現有圖像的列表。

參數：

**image\_status** (必要)

依影像狀態篩選。

有效值：AVAILABLE | PENDING | FAILED

**region**

列出內置於給定的圖像AWS 區域。

**next\_token**

用於分頁請求的令牌。

## build\_image

```
build_image(image_configuration, image_id, suppress_validators,  
            validation_failure_level, dry_run, rollback_on_failure, region)
```

建立自訂AWS ParallelCluster給定區域中的圖像。

參數：

### **image\_configuration** (必要)

將圖像配置為 Python 數據。

### **image\_id** (必要)

影像識別碼。

### **suppress\_validators**

識別要隱藏的一或多個映像組態驗證程式。

格式：(ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

造成映像建立失敗的最低驗證層級。預設值為 ERROR。

有效值：INFO | WARNING | ERROR

### **dry\_run**

如果設定為True,AWS ParallelCluster在不建立任何資源的情況下執行要求驗證。您可以使用它來驗證映像配置。預設值為 False。

### **rollback\_on\_failure**

如果設定為True,AWS ParallelCluster在失敗時自動啟動映像堆疊復原。預設值為 False。

### **region**

該圖像AWS 區域。

## delete\_image

```
delete_image(image_id, region, force)
```

刪除給定區域中的圖像。

參數：

**image\_id** (必要)

影像識別碼。

**region**

該圖像AWS 區域。

**force**

如果設定為True,AWS ParallelCluster如果執行個體正在使用 AMI 或共用 AMI ，則強制刪除。預設值為 False。

## describe\_image

```
describe_image(image_id, region)
```

取得有關現有映像檔的詳細資訊。

參數：

**image\_id** (必要)

影像識別碼。

**region**

該圖像AWS 區域。

## 影像和堆疊記錄作業

主題

- [list\\_image\\_log\\_streams](#)
- [get\\_image\\_log\\_events](#)
- [get\\_image\\_stack\\_events](#)
- [list\\_official\\_images](#)

## list\_image\_log\_streams

```
list_image_log_streams(image_id, region, next_token)
```

列出映像檔的記錄資料流。

參數：

### **image\_id** (必要)

影像識別碼。

### **region**

該圖像AWS 區域。

### **next\_token**

用於分頁請求的令牌。

## get\_image\_log\_events

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head,
limit, start_time, end_time)
```

獲取給定圖像和日誌流的日誌事件。

參數：

### **image\_id** (必要)

影像識別碼。

### **log\_stream\_name** (必要)

記錄資料流名稱。

### **region**

該圖像AWS 區域。

### **next\_token**

用於分頁請求的令牌。

## **start\_from\_head**

如果設定為True,AWS ParallelCluster首先傳回最早的記錄事件。如果設定為False , 它會先傳回最新的記錄事件。預設值為 False。

## **limit**

傳回的記錄事件數目上限。如果未指定值 , 則最多可容納 1 MB 回應大小的記錄檔數目 , 最多 10,000 個記錄事件。

## **start\_time**

記錄事件的時間範圍開始 , 以 ISO 8601 格式表示 ; 例如 , '2021-01-01T20:00:00Z'。包含時間戳記等於或晚於此時間的事件。

## **end\_time**

記錄事件的時間範圍結束時間 , 以 ISO 8601 格式表示 ; 例如 , '2021-01-01T20:00:00Z'。不會包含時間戳記等於或晚於此時間的事件。

## **get\_image\_stack\_events**

```
get_image_stack_events(image_id, region, next_token)
```

獲取給定圖像的堆棧事件。

參數 :

### **image\_id** (必要)

影像識別碼。

### **region**

該圖像AWS 區域。

### **next\_token**

用於分頁請求的令牌。

## **list\_official\_images**

```
list_official_images(region,os, architecture)
```



檢索官員名單AWS ParallelCluster圖像。

參數：

### region

該圖像AWS 區域。

### os

依作業系統分佈進行篩選。預設值為無篩選。

### architecture

依架構篩選。預設值為無篩選。

## 範例

主題

- [建立 叢集](#)

### 建立 叢集

當您運行下面的示例腳本，與存儲在您的環境中給定的輸入，你創建一個集群。叢集配置是以 Python 資料類型為基礎建立[叢集配置文件](#)。

```
import os
import pprint
import pcluster.lib as pc
pp = pprint.PrettyPrinter()

HEAD_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
COMPUTE_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
KEY_NAME = os.environ["KEY_NAME"]
CONFIG = {'Image': {'Os': 'alinux2'},
          'HeadNode': {'InstanceType': 't2.large',
                       'Networking': {'SubnetId': HEAD_NODE_SUBNET},
                       'Ssh': {'KeyName': KEY_NAME}},
          'Scheduling': {'Scheduler': 'slurm',
                         'SlurmQueues':
                         [{'Name': 'queue0',
```

```

    'ComputeResources':
      [{ 'Name': 'queue0-i0', 'InstanceType': 't2.micro',
        'MinCount': 0, 'MaxCount': 10}],
    'Networking': { 'SubnetIds': [COMPUTE_NODE_SUBNET]}}}}

```

```
pp.pprint(pc.create_cluster(cluster_name="mycluster", cluster_configuration=CONFIG))
```

輸出：

```

{'cluster': {'cloudformationStackArn': 'arn:aws:cloudformation:us-
east-2:123456789012:stack/mycluster/00000000-aaaa-1111-999-000000000000',
  'cloudformationStackStatus': 'CREATE_IN_PROGRESS',
  'clusterName': 'mycluster',
  'clusterStatus': 'CREATE_IN_PROGRESS',
  'region': 'us-east-2',
  'scheduler': {'type': 'slurm'},
  'version': '3.7.0'}}

```

## AWS Lambda為了AWS ParallelCluster蟒蛇圖書館

您可以部署 Lambda 層和執行階段以存取AWS ParallelCluster蟒蛇庫。我們主辦AWS ParallelCluster您可以透過輸入 zip 檔案連結來使用的 zip 檔案，如下列步驟所述。Lambda 會使用 zip 檔案來準備執行階段環境，以支援對 Python 程式庫的存取。該AWS ParallelClusterPython 庫被添加了AWS ParallelCluster版本 3.5.0。您只能將程式庫用於版本 3.5.0 及更新版本。

託管的 zip 文件網址的格式如下：`s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`

### 開始存取AWS ParallelCluster與蟒蛇庫AWS Lambda

創建一個拉姆達圖層

1. 登入AWS Management Console並導航到AWS Lambda控制台。
2. 在導覽窗格中，選取圖層，然後建立圖層。
3. 輸入圖層的名稱，然後選擇從亞馬遜 S3 上傳文件。
4. 輸入壓縮文件的網址：`s3://aws-region-id-aws-並行群集/並行群集/3.7.0/層/aws-並行群集/羊肉圖層.zip`。
5. 對於相容架構，選擇X86\_64體系結構。

6. 對於相容的執行階段，選擇蟒蛇 3.9執行階段。
7. 選擇 建立。

### 使用您的 Lambda 層

1. 在 Lambda 主控台導覽窗格中，選取函數，然後建立函數。
2. 輸入函數的名稱。
3. 對於运行时，選擇蟒蛇 3.9執行階段。
4. 對於建築，選擇X86\_64體系結構。
5. 選擇 建立函數。
6. 建立函數後，選擇圖層並選擇新增圖層。
7. 选择自訂圖層，然後選擇您在之前步驟中建立的圖層。
8. 選擇圖層版本。
9. 選擇 Add (新增)。
10. 您的 Lambda 需要許可才能管理使用以下方式AWS ParallelCluster。使用中列出的權限建立 Lambda 角色[基本 AWS ParallelCluster pcluster使用者原則](#)。

您現在可以訪問AWS ParallelCluster從 Python 庫中描述[AWS ParallelClusterPython 函式庫](#)。

# 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服務來建置 HPC 環境的參考。

## 主題

- [AWS ParallelCluster 程序](#)
- [AWS 使用的服務 AWS ParallelCluster](#)
- [AWS ParallelCluster內部目錄](#)

## AWS ParallelCluster 程序

本節適用於使用部署的叢集Slurm。與此排程器搭配使用時，可透過與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。

## clustermgtd

叢集管理常駐程式會執行下列工作。

- 非作用中分割區清理
- 管理與容量區塊相關聯的Slurm保留區和節點 (請參閱下節)
- 靜態容量管理：確保靜態容量始終保持正常運作
- 與 Amazon 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 Amazon EC2 實例管理 ( Amazon EC2 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)

## 管理與容量區塊相關聯的Slurm保留區和節點

ParallelCluster 支援 Machine Learning (CB) 的隨需容量保留 (ODCR) 和容量區塊。與 ODCR 不同，CB 可以擁有 future 的開始時間，並且具有時間限制。

Clustermgtd 會在迴圈中搜尋運作狀態不良的節點，並終止任何關閉的 Amazon EC2 執行個體，如果它們是靜態節點，則以新執行個體取代它們。

ParallelCluster 管理與容量區塊相關聯的靜態節點不同。AWS ParallelCluster 即使 CB 尚未處於作用中狀態，仍會建立叢集，而當 CB 處於作用中狀態時，執行個體也會自動啟動。

與尚未使用中的 CBS 相關聯的計算資源對應的Slurm節點會保持維護狀態，直到到達 CB 開始時間為止。Slurm節點將保持與Slurm管理員使用者相關聯的保留/維護狀態，這表示他們可以接受工作，但在移除保留項目之前，工作將保留處於擱置狀態。Slurm

Clustermgtd 會自動建立/刪除保Slurm留區，並根據 CB 狀態將相關 CB 節點置於維護中。CB 處於作用中狀態時，將移除Slurm保留區、節點將啟動，並可供擱置的工作或新工作提交使用。

到達 CB 結束時間時，節點會移回保留/維護狀態。當 CB 不再處於作用中狀態且執行個體終止時，使用者可以將工作重新提交/重新報告至新佇列/運算資源。

## clusterstatusmgtd

叢集狀態管理常駐程式會管理運算叢集狀態更新。每分鐘都會擷取 DynamoDB 表格中儲存的叢集狀態，並管理任何停止/啟動請求。

## computemgtd

計算管理常駐程式 (computemgtd) 處理序會在每個叢集運算節點上執行。計算管理常駐程式每五 (5) 分鐘就會確認頭節點可以連線且狀態良好。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 使用的服務 AWS ParallelCluster

以下 Amazon Web Services ( AWS ) 服務被使用 AWS ParallelCluster。

主題

- [Amazon API Gateway](#)
- [AWS Batch](#)

- [AWS CloudFormation](#)
- [Amazon CloudWatch](#)
- [Amazon CloudWatch 活動](#)
- [Amazon CloudWatch 日誌](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx for Lustre](#)
- [Amazon FSx NetApp](#)
- [Amazon FSx for OpenZFS](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon RDS](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service](#)
- [Amazon Simple Storage Service](#)
- [Amazon VPC](#)
- [Elastic Fabric Adapter](#)
- [EC2 Image Builder](#)
- [NICE DCV](#)

## Amazon API Gateway

Amazon API Gateway 是一種 AWS 用於建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API 的服務

AWS ParallelCluster 使用 API Gateway 來託管 AWS ParallelCluster API。

如需有關的詳細資訊 AWS Batch，請參閱 <https://aws.amazon.com/api-gateway/> 和 <https://docs.aws.amazon.com/apigateway/>。

## AWS Batch

AWS Batch 是 AWS 受管理的工作排程器服務。它會在 AWS Batch 叢集中動態佈建最佳數量和運算資源類型 (例如 CPU 或記憶體最佳化執行個體)。這些資源是根據批次工作的特定需求 (包括磁碟區需求) 佈建。有了 AWS Batch，您不需要安裝或管理額外的批次運算軟體或伺服器叢集，就能有效執行工作。

AWS Batch 僅與 AWS Batch 叢集搭配使用。

如需有關的詳細資訊 AWS Batch，請參閱 <https://aws.amazon.com/batch/> 和 <https://docs.aws.amazon.com/batch/>。

## AWS CloudFormation

AWS CloudFormation 是一種提供通用語言的 infrastructure-as-code 服務，用於在雲端環境中建模 AWS 和佈建協力廠商應用程式資源。它是使用的主要服務 AWS ParallelCluster。中 AWS ParallelCluster 的每個叢集都以堆疊表示，而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 命令直接對應於 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對啟動叢集的 AWS CloudFormation AWS 區域端點進行 HTTPS 呼叫。

如需有關的詳細資訊 AWS CloudFormation，請參閱 <https://aws.amazon.com/cloudformation/> 和 <https://docs.aws.amazon.com/cloudformation/>。

## Amazon CloudWatch

Amazon CloudWatch (CloudWatch) 是一項監控和可觀察性服務，可為您提供資料和可操作的見解。這些見解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。中 AWS ParallelCluster，用 CloudWatch 於儀表板，用於監視和記錄 Docker 映像構建步驟以及 AWS Batch 作業的輸出。

在 AWS ParallelCluster 版本 2.10.0 之前，僅用 CloudWatch 於叢集。AWS Batch

如需有關的詳細資訊 CloudWatch，請參閱 <https://aws.amazon.com/cloudwatch/> 和 <https://docs.aws.amazon.com/cloudwatch/>。

## Amazon CloudWatch 活動

亞馬遜 CloudWatch 活動 (CloudWatch 事件) 提供近乎即時的系統事件串流，這些事件描述了 Amazon Web 服務 (AWS) 資源中的變更。使用您可以快速設定的簡單規則，您可以比對事件並將它們路由到一或多個目標函數或串流。在中 AWS ParallelCluster，CloudWatch 事件用於 AWS Batch 工作。

如需有關 CloudWatch 活動的詳細資訊，請參閱 <https://docs.aws.amazon.com//eventbridge/latest/userguide/eb-cwe-now-eb>。

## Amazon CloudWatch 日誌

Amazon CloudWatch 日誌 ( CloudWatch 日誌 ) 是 Amazon 的核心功能之一 CloudWatch。您可以使用它來監視、儲存、檢視和搜尋記錄檔中使用的許多元件 AWS ParallelCluster。

在 2.6.0 AWS ParallelCluster 版之前，CloudWatch 記錄僅用於 AWS Batch 叢集。

如需詳細資訊，請參閱 [與 Amazon Amazon Amazon CloudWatch 的整合](#)。

## AWS CodeBuild

AWS CodeBuild (CodeBuild) 是 AWS 受管理的持續整合服務，可符合原始程式碼、執行測試，以及產生可供部署的軟體套件。在中 AWS ParallelCluster，可 CodeBuild 用來在建立叢集時自動且透明地建置 Docker 映像。

CodeBuild 僅與 AWS Batch 叢集搭配使用。

如需有關的詳細資訊 CodeBuild，請參閱 <https://aws.amazon.com/codebuild/> 和 <https://docs.aws.amazon.com/codebuild/>。

## Amazon DynamoDB

Amazon DynamoDB 料庫是一種快速且靈活的 NoSQL 資料庫服務。它是用來存儲群集的最小狀態信息。頭節點會追蹤 DynamoDB 表格中佈建的執行個體。

DynamoDB 不會與叢集搭配 AWS Batch 使用。

如需有關 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續性儲存。所有 Amazon EBS 設定都可以透過組態傳遞。Amazon EBS 磁碟區可以是空的，也可以從現有的 Amazon EBS 快照初始化。

有關 Amazon EBS 的更多信息，請參閱 <https://aws.amazon.com/ebs/> 和 <https://docs.aws.amazon.com/ebs/>。



## Amazon Elastic Compute Cloud

Amazon 彈性運算雲 ( Amazon EC2 ) 提供的計算容量 AWS ParallelCluster。主節點和運算節點是 Amazon EC2 執行個體。可選取支援 HVM 的任何執行個體類型。頭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，部分或全部運算節點也可以作為競價型執行個體啟動。在執行個體上找到的執行個體存放區磁碟區會掛載為條帶 LVM 磁碟區。

如需有關 Amazon EC2 的詳細資訊，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) 是全受管的 Docker 容器登錄，可讓您輕鬆存放、管理和部署 Docker 容器映像。在中 AWS ParallelCluster，Amazon ECR 會儲存建立叢集時建立的泊塢視窗映像。然後會使用 Docker 映像檔 AWS Batch 來執行已提交工作的容器。

Amazon ECR 僅適用於 AWS Batch 叢集。

如需更多資訊，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、可擴展且全受管的彈性 NFS 檔案系統，可與 AWS 雲端 服務和現場部署資源搭配使用。如果 [EfsSettings](#) 有指定，就會使用 Amazon EFS。在 2.1.0 AWS ParallelCluster 版中新增了對 Amazon EFS 的 Support 援。

如需有關 Amazon EFS 的詳細資訊，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。指定 FSx for Lustre 會在指定 [FsxLustreSettings](#) 屬性時使用。在 2.2.1 版中 AWS ParallelCluster 增加了 FSx for Lustre 的 Support。

如需 FSx for Lustre 的更多資訊，請參閱 <https://aws.amazon.com/fsx/lustre/> 和 <https://docs.aws.amazon.com/fsx/>。

## Amazon FSx NetApp

FSx for ONTAP 提供以熱門 ONTAP 檔案系統為基礎 NetApp 的全受管共用儲存系統。

如 [FsxOntapSettings](#) 屬性果有指定，則會使用 ONTAP 的 FSx。在 3.2.0 AWS ParallelCluster 版中新增了對 FSx 的 Support 援。

如需有關 FSx 適用於 ONTAP 的詳細資訊，請參閱 <https://aws.amazon.com/fsx/netapp-ontap/> 和 <https://docs.aws.amazon.com/fsx/>。

## Amazon FSx for OpenZFS

適用於 OpenZFS 的 FSx 提供建置在熱門 OpenZFS 檔案系統上的完全受管理的共用儲存系統。如果有指 [FsxOpenZfsSettings](#) 屬性定，就會使用 OpenZF 的 FSx。在 AWS ParallelCluster 3.2.0 版中新增了對 FSx 的 Support 援。

如需有關開放式空間轉換器 FSx 的詳細資訊，請參閱 <https://aws.amazon.com/fsx/openzfs/> 和 <https://docs.aws.amazon.com/fsx/>。

## AWS Identity and Access Management

AWS Identity and Access Management (IAM) 用於 AWS ParallelCluster 為每個個別叢集特有的執行個體提供 Amazon EC2 的最低權限 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。

透過 AWS Batch 叢集，在建立叢集時，也會為與 Docker 映像建立程序相關的元件建立 IAM 角色。這些元件包括允許在 Amazon ECR 儲存庫中新增和刪除 Docker 映像檔的 Lambda 函數。它們還包括允許刪除針對叢集和 CodeBuild 專案建立之 Amazon S3 儲存貯體的功能。AWS Batch 資源、執行個體和工作也有角色。

如需 IAM 的詳細資訊，請參閱 <https://aws.amazon.com/iam/> 和 <https://docs.aws.amazon.com/iam/>。

## AWS Lambda

AWS Lambda (Lambda) 執行可協調建立碼頭影像的函數。Lambda 也會管理自訂叢集資源的清理工作，例如存放在 Amazon ECR 儲存庫和 Amazon S3 上的 Docker 映像檔。

如需有關 Lambda 的更多資訊，請參閱 <https://aws.amazon.com/lambda/> 和 <https://docs.aws.amazon.com/lambda/>。

## Amazon RDS

Amazon Relational Database Service (Amazon RDS) 是一種 Web 服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。

AWS ParallelCluster 使用 Amazon RDS AWS Batch 和 Slurm。

有關 Amazon RDS 的更多信息，請參閱 <https://aws.amazon.com/rds/> 和 <https://docs.aws.amazon.com/rds/>。

## Amazon Route 53

Amazon Route 53 (Route 53) 用於為每個運算節點建立具有主機名稱和完整網域名稱的託管區域。

如需 53 號公路的詳細資訊，請參閱 <https://aws.amazon.com/route53/> 和 <https://docs.aws.amazon.com/route53/>。

## Amazon Simple Notification Service

(Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。

AWS ParallelCluster 使用 Amazon SNS 進行 API 託管。

有關 Amazon SNS 的更多信息，請參閱 <https://aws.amazon.com/sns/> 和 <https://docs.aws.amazon.com/sns/>。

## Amazon Simple Storage Service

亞馬遜簡單儲存服務 (Amazon S3) 會將 AWS ParallelCluster 範本存放在每個範本中 AWS 區域。

AWS ParallelCluster 可以設定為允許 CLI /SDK 工具使用 Amazon S3。

AWS ParallelCluster 也會在您的叢集中建立 Amazon S3 儲存貯體，AWS 帳戶以存放叢集所使用的資源，例如叢集組態檔。AWS ParallelCluster 在您建立叢集的每個 AWS 區域 儲存貯體中維護一個 Amazon S3 儲存貯體。

使用 AWS Batch 叢集時，帳戶中會使用 Amazon S3 儲存貯體來存放相關資料。例如，儲存貯體會儲存從提交的工作建立 Docker 映像檔和指令碼時所建立的人工因素。

如需更多資訊，請參閱 <https://aws.amazon.com/s3/> 和 <https://docs.aws.amazon.com/s3/>。

## Amazon VPC

Amazon VPC 會定義叢集中節點所使用的網路。

有關 Amazon VPC 的更多信息，請參閱 <https://aws.amazon.com/vpc/> 和 <https://docs.aws.amazon.com/vpc/>。

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) 是一種網路介面，適用於執行個體，客戶可用來執行需要大規模高階節點間通訊的應用程式。AWS

如需 EC2 Image Builder 的詳細資訊，請參閱 <https://aws.amazon.com/hpc/efa/>。

## EC2 Image Builder

EC2 Image Builder 是一項全受管 AWS 服務，可協助您自動化自訂、安全和 up-to-date 伺服器映像的建立、管理和部署。

AWS ParallelCluster 使用 Image Builder 來建立和管理影 AWS ParallelCluster 像。

如需 EC2 Image Builder 的詳細資訊，請參閱 <https://aws.amazon.com/image-builder/> 和 <https://docs.aws.amazon.com/imagebuilder/>。

## NICE DCV

NICE DCV 是一種高效能遠端顯示通訊協定，可透過不同網路條件，將遠端桌面平台和應用程式串流傳送至任何裝置的安全方式。當指定 [HeadNode 區段/Dcv](#) 設置時，使用 NICE DCV DCV。在 2.5.0 AWS ParallelCluster 版本中添加了對 NICE DCV DCV 的 Support。

有關 NICE DCV DCV 的更多信息，請參閱 <https://aws.amazon.com/hpc/dcv/> 和 <https://docs.aws.amazon.com/dcv/>。

## AWS ParallelCluster內部目錄

有數個內部目錄AWS ParallelCluster用於在叢集內共用資料。在頭節點、計算節點和登入節點之間共用下列目錄：

/opt/slurm

`/opt/intel`

`/opt/parallelcluster/shared` (only with compute nodes)

`/opt/parallelcluster/shared_login_nodes` (only with login nodes)

`/home` (unless specified in `SharedStorage`)

#### Note

依預設，這些目錄會建立在頭節點 EBS 磁碟區上，並以 NFS 匯出的形式共用至運算和登入節點。從 AWS ParallelCluster 3.8 開始，您可以透過 AWS ParallelCluster 將 [SharedStorageType](#) 參數設定為 `efs` 來建立和管理 Amazon EFS 檔案系統來託管和共用這些直接點。

當叢集向外延展時，透過 EBS 磁碟區進行 NFS 匯出可能會造成效能瓶頸。使用 EFS，您可以在叢集向外擴充時避免 NFS 匯出，並避免與它們相關的效能瓶頸。

# 教學課程

下列教學課程說明如何開始使用第 3 AWS ParallelCluster 版，並提供一些常見工作的最佳實務指引。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需更多詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

## 主題

- [執行您的第一份工作 AWS ParallelCluster](#)
- [構建自定義 AWS ParallelCluster AMI](#)
- [整合作用中目錄](#)
- [使用 AWS KMS 金鑰設定共用儲存區加密](#)
- [在多佇列模式叢集中執行工作](#)
- [應用 AWS ParallelCluster 程式介面](#)
- [使用Slurm會計建立叢集](#)
- [使用外部Slurmdbd帳戶建立叢集](#)
- [還原至先前的AWS系統管理員文件版本](#)
- [建立叢集 AWS CloudFormation](#)
- [使用地形部署 ParallelCluster API](#)
- [使用地形建立叢集](#)
- [使用地形創建自定義 AMI](#)
- [AWS ParallelCluster與身分識別中心的 UI 整合](#)

## 執行您的第一份工作 AWS ParallelCluster

本教程將引導您完成運行您的第一個 Hello World 工作 AWS ParallelCluster

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

## 必要條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 `pcluster` CLI 所需[權限](#)的 IAM 角色。

## 確認安裝

首先，我們驗證 AWS ParallelCluster 是否正確，包括 Node.js 依賴項，已安裝和配置。

```
$ node --version
v16.8.0
$ pcluster version
{
  "version": "3.7.0"
}
```

這會傳回的執行版本 AWS ParallelCluster。

## 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-cluster \
  --cluster-name hello-world \
  --cluster-configuration hello-world.yaml
```

### Note

必須為大多數指 `pcluster` 令指定 AWS 區域 要使用的。如果未在 `AWS_DEFAULT_REGION` 環境變數或 `~/.aws/config` 檔案 [default] 區段中指定 `region` 設定，則必須在指 `pcluster` 令行上提供 `--region` 參數。

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 --config hello-world.yaml
```

如果命 `pcluster create-cluster` 令成功，您會看到類似下列內容的輸出：

```
{
  "cluster": {
    "clusterName": "hello-world",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:xxx:stack/xxx",
    "region": "...",
    "version": "...",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

您可以使用下列方式監視叢集的建立：

```
$ pcluster describe-cluster --cluster-name hello-world
```

建立叢集時的 `clusterStatus` 報告 `CREATE_IN_PROGRESS`。成功建立叢集時 `clusterStatus` 轉換為 `CREATE_COMPLETE`。輸出還為我們提供了 `publicIpAddress` 和我們 `privateIpAddress` 的頭節點。

## 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 --cluster-name hello-world -i /path/to/keyfile.pem
```

登入之後，執行命令 `sinfo` 來驗證您的運算節點是否已設置和設定。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   10   idle~ queue1-dy-queue1t2micro-[1-10]
```

輸出顯示，我們有一個隊列在我們的集群，最多十個節點。



## 使用思慮姆運行你的第一份工作

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 `sbatch` 來提交任務，並驗證它是否執行。

```
$ sbatch hellojob.sh
Submitted batch job 2
```

現在，您可以檢視佇列並檢查此任務的狀態。新 Amazon EC2 執行個體的佈建會在背景啟動。您可以使用 `sinfo` 指令監視叢集執行個體的状态。

```
$ squeue
      JOBID PARTITION    NAME     USER ST       TIME  NODES NODELIST(REASON)
         2     queue1 hellojob ec2-user CF       3:30      1 queue1-dy-
queue1t2micro-1
```

輸出顯示工作已提交至 `queue1`。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 57 Sep  1 14:25 hellojob.sh
-rw-rw-r-- 1 ec2-user ec2-user 43 Sep  1 14:30 slurm-2.out
```

在輸出中，我們看到一個 "out" 文件。我們可以從我們的工作中看到輸出：

```
$ cat slurm-2.out
Hello World from queue1-dy-queue1t2micro-1
```

輸出也會顯示我們的任務已在執行個體 `queue1-dy-queue1t2micro-1` 上成功執行。

在您剛剛建立的叢集中，叢集的所有節點之間只會共用主目錄。

若要進一步瞭解如何建立和使用叢集，請參閱[最佳實務](#)。

如果您的應用程式需要共用軟體、程式庫或資料，請考慮下列選項：

- 如中所述，建立包含您軟體的 AWS ParallelCluster 已啟用自訂 AMI [構建自定義 AWS ParallelCluster AMI](#)。
- 使用 AWS ParallelCluster 組態檔案中的 [StorageSettings](#) 選項來指定共用檔案系統，並將已安裝的軟體儲存在指定的掛載位置。
- 用 [自訂引導動作](#) 於自動執行叢集每個節點的啟動程序。

## 構建自定義 AWS ParallelCluster AMI

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

### Important

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 [自訂引導動作](#) 本節。判斷您要進行的修改是否可以編寫指令碼，並在 future 的 AWS ParallelCluster 版本中提供支援。

雖然一般來說建立自訂 AMI 並不理想，但在某些特定情況下，建置自訂 AMI AWS ParallelCluster 是必要的。本教程介紹了如何為這些案例構建自定義 AMI。

### 必要條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 [pcluster](#) CLI 和建置映像所需 [權限](#) 的 IAM 角色。

## 如何自定義 AWS ParallelCluster AMI

有兩種方法可以構建自定義 AWS ParallelCluster AMI。這兩種方法之一是使用 AWS ParallelCluster CLI 構建新的 AMI。另一種方法要求您進行手動修改以構建 AWS 帳戶。

### 建立自訂 AWS ParallelCluster AMI

如果您有自定義 AMI 和軟件，則可以在其 AWS ParallelCluster 上應用所需的更改。AWS ParallelCluster 依賴 EC2 Image Builder 服務來建置自訂 AMI。如需詳細資訊，請參閱 [Image Builder 使用指南](#)。

關鍵點：

- 該過程大約需要 1 小時。如果在構建時 [Components](#) 要安裝其他 [Build/](#)，則此時間可能會有所不同。
- AMI 被標記為主要組件的版本。其中包括核心、排程器和 [EFA 驅動程式](#)。AMI 說明中也會報告元件版本的子集。
- 從 AWS ParallelCluster 3.0.0 開始，可以使用一組新的 CLI 命令來管理映像的生命週期。其中包括 [build-image](#)、[list-images](#)、[describe-image](#) 和 [delete-image](#)。
- 此方法是可重複的。您可以重新執行它以保持 AMI 更新 (例如作業系統更新)，然後在更新現有叢集時使用它們。

#### Note

如果在 AWS 中國分區中使用此方法，則可能會收到網絡錯誤。例如，當 `pcluster build-image` 命令從作業系統儲存庫 GitHub 或從作業系統儲存庫下載套件時，您可能會看到這些錯誤。如果發生這種情況，我們建議您使用下列其中一種替代方法：

1. 遵循繞過此命令的 [修改 AWS ParallelCluster AMI](#) 方法。
2. 在另一個分割區和區域中建立映像檔，例如 `us-east-1`，然後儲存還原影像，以便將其移至中國區域。如需詳細資訊，請參閱 [Amazon EC2 使用者指南中的使用 S3 存放和還原 AMI](#)。

步驟：

1. 設定您的 AWS 帳戶 認證，以便 AWS ParallelCluster 用戶端可以代表您呼叫 AWS API 作業。如需必要許可的清單，請參閱 [AWS Identity and Access Management 權限 AWS ParallelCluster](#)。

2. 創建一個基本的構建映像配置文件。若要執行此操作，請指定要用來建置映像檔的 [ParentImage](#)。 [InstanceType](#) 這些被用作創建 AMI 的起點。有關可選構建參數的詳細信息，請參閱 [映像配置](#)。

```
Build:
  InstanceType: <BUILD_INSTANCE_TYPE>
  ParentImage: <BASE_AMI_ID>
```

3. 使用 CLI 命令 [pcluster build-image](#) 從您提供做為基礎的 AMI 開始建立 AMI。AWS ParallelCluster

```
$ pcluster build-image --image-id IMAGE_ID --image-configuration IMAGE_CONFIG.yaml --
region REGION
{
  "image": {
    "imageId": "IMAGE_ID",
    "imageBuildStatus": "BUILD_IN_PROGRESS",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
    "region": "us-east-1",
    "version": "3.7.0"
  }
}
```

#### Warning

`pcluster build-image` 使用預設的 VPC。如果您使用 AWS Control Tower 或 AWS 登陸區域刪除預設 VPC，則必須在映像設定檔中指定子網路 ID。如需詳細資訊，請參閱 [SubnetId](#)。

如需其他參數的清單，請參閱命 [pcluster build-image](#) 令參考頁面。上述命令的結果如下：

- 會根據映像組態建立 CloudFormation 堆疊。堆疊包含組建所需的所有 EC2 Image Builder 資源。
- 創建的資源包括可以添加自定義 Image Bu AWS ParallelCluster ilder 組件的官方 Image Builder 組件。若要瞭解如何建立自訂元件，請參閱適用於公共部門客戶的 HPC 研討會中的 [自訂 AMI 範例](#)。

- EC2 Image Builder 會啟動建置執行個體、套用 AWS ParallelCluster 食譜、安裝 AWS ParallelCluster 軟體堆疊，以及執行必要的設定工作。AWS ParallelCluster 食譜用於構建和引導 AWS ParallelCluster。
  - 執行個體會停止，並從中建立新 AMI。
  - 另一個實例是從新創建的 AMI 啟動的。在測試階段，EC2 Image Builder 會執行映像產生器元件中定義的測試。
  - 如果構建成功，堆棧被刪除。如果組建失敗，則會保留堆疊並可供檢查。
4. 您可以執行下列命令來監視建置程序的狀態。構建完成後，您可以運行它以檢索響應中給出的 AMI ID。

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION

# BEFORE COMPLETE
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-
delete.s3.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/
configs/image-config.yaml?...",
  },
  "imageId": "IMAGE_ID",
  "imagebuilderImageStatus": "BUILDING",
  "imageBuildStatus": "BUILD_IN_PROGRESS",
  "cloudformationStackStatus": "CREATE_IN_PROGRESS",
  "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
  "region": "us-east-1",
  "version": "3.7.0",
  "cloudformationStackTags": [
    {
      "value": "3.7.0",
      "key": "parallelcluster:version"
    },
    {
      "value": "IMAGE_ID",
      "key": "parallelcluster:image_name"
    },
    ...
  ],
  "imageBuildLogsArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-IMAGE_ID",
  "cloudformationStackCreationTime": "2022-04-05T21:36:26.176Z"
}
```

```
# AFTER COMPLETE
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/image-config.yaml?Signature=..."
  },
  "imageId": "IMAGE_ID",
  "imageBuildStatus": "BUILD_COMPLETE",
  "region": "us-east-1",
  "ec2AmiInfo": {
    "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
    "amiId": "ami-1234stuv5678wxyz",
    "description": "AWS ParallelCluster AMI for alinux2, kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64, efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.e17.x86_64, slurm-20-11-8-1",
    "state": "AVAILABLE",
    "tags": [
      {
        "value": "2021.3.11591-1.e17.x86_64",
        "key": "parallelcluster:dcv_version"
      },
      ...
    ],
    "architecture": "x86_64"
  },
  "version": "3.7.0"
}
```

5. 若要建立叢集，請在叢集配置的 [CustomAmi](#) 欄位中輸入 AMI ID。

## 疑難排解和監控 AMI 建立程序

映像建立會在大約一小時內完成。您可以透過執行 [pcluster describe-image](#) 命令或記錄擷取命令來監視程序。

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

該 [build-image](#) 命令會建立一個 CloudFormation 堆疊，其中包含建置映像所需的所有 Amazon EC2 資源，並啟動 EC2 Image Builder 程序。

運行命 [build-image](#) 令後，可以通過使用檢索 CloudFormation 堆棧事件 [pcluster get-image-stack-events](#)。您可以使用 `--query` 參數篩選結果以查看最新事件。如需詳細資訊，請參閱《使用指南》中的 AWS Command Line Interface [〈篩選 AWS CLI 輸出〉](#)。

```
$ pcluster get-image-stack-events --image-id IMAGE_ID --region REGION --query
"events[0]"
{
  "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
  "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-IMAGE_ID/3.7.0/1",
  "resourceStatus": "CREATE_IN_PROGRESS",
  "resourceStatusReason": "Resource creation Initiated",
  "resourceProperties": "{\"InfrastructureConfigurationArn\":
\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-abcd1234-ef56-gh78-ij90-1234abcd5678\", \"ImageRecipeArn\":
\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/parallelclusterimage-
IMAGE_ID/3.7.0\", \"DistributionConfigurationArn\": \"arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-abcd1234-ef56-
gh78-ij90-1234abcd5678\", \"Tags\": {\"parallelcluster:image_name\": \"IMAGE_ID\",
\"parallelcluster:image_id\": \"IMAGE_ID\"}}",
  "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/IMAGE_ID/abcd1234-
ef56-gh78-ij90-1234abcd5678",
  "stackName": "IMAGE_ID",
  "logicalResourceId": "ParallelClusterImage",
  "resourceType": "AWS::ImageBuilder::Image",
  "timestamp": "2022-04-05T21:39:24.725Z"
}
```

大約 15 分鐘後，堆疊事件會顯示在與 Image Builder 建立相關的記錄事件項目中。您現在可以使 [pcluster list-image-log-streams](#) 用和指令列出影像記錄串流並監視映像產生器步 [pcluster get-image-log-events](#) 驟。

```
$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
--query 'logStreams[*].logStreamName'

"3.7.0/1"
]

$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
--log-stream-name 3.7.0/1 --limit 3
{
  "nextToken": "f/36295977202298886557255241372854078762600452615936671762",
```

```

"prevToken": "b/36295977196879805474012299949460899222346900769983430672",
"events": [
  {
    "message": "ExecuteBash: FINISHED EXECUTION",
    "timestamp": "2022-04-05T22:13:26.633Z"
  },
  {
    "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-abcd1234-ef56-gh78-ij90-1234abcd5678/3.7.0/1",
    "timestamp": "2022-04-05T22:13:26.741Z"
  },
  {
    "message": "TOE has completed execution successfully",
    "timestamp": "2022-04-05T22:13:26.819Z"
  }
]
}

```

繼續使用[describe-image](#)指令檢查，直到看到狀BUILD\_COMPLETE態為止。

```

$ pcluster describe-image --image-id IMAGE_ID --region REGION
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
  },
  "imageId": "IMAGE_ID",
  "imageBuildStatus": "BUILD_COMPLETE",
  "region": "us-east-1",
  "ec2AmiInfo": {
    "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
    "amiId": "ami-1234stuv5678wxyz",
    "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
    "state": "AVAILABLE",
    "tags": [
      {
        "value": "2021.3.11591-1.el7.x86_64",
        "key": "parallelcluster:dcv_version"
      },
      ...
    ]
  }
}

```



```
    ],  
    "architecture": "x86_64"  
  },  
  "version": "3.7.0"  
}
```

如果您需要疑難排解自訂 AMI 建立問題，請依照下列步驟所述建立映像記錄的封存檔。

視 `--output` 參數而定，可以將日誌存檔到 Amazon S3 儲存貯體或本機檔案中。

```
$ pcluster export-image-logs --image-id IMAGE_ID --region REGION \  
--bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER \  
{  
  "url": "https://BUCKET_NAME.s3.us-east-1.amazonaws.com/BUCKET-FOLDER/IMAGE_ID-  
logs-202209071136.tar.gz?AWSAccessKeyId=..."  
}  
  
$ pcluster export-image-logs --image-id IMAGE_ID \  
--region REGION --bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER --output-file /tmp/  
archive.tar.gz  
{  
  "path": "/tmp/archive.tar.gz"  
}
```

封存包含與 Image Builder 處理作業和 AWS CloudFormation 堆疊事件相關的 CloudWatch 記錄資料流。該命令可能需要幾分鐘的時間才能運行。

## 管理自訂 AMI

從 AWS ParallelCluster 3.0.0 開始，CLI 中已新增一組新的指令來建置、監視和管理映像生命週期。如需有關指令的詳細資訊，請參閱 [pcluster 指令](#)。

## 修改 AWS ParallelCluster AMI

此方法包括通過在其上添加自定義來修改官方 AWS ParallelCluster AMI。基礎 AWS ParallelCluster AMI 會以新版本進行更新。這些 AMI 具有在安裝和設定時運 AWS ParallelCluster 作所需的所有元件。您可以從其中之一開始作為您的基地。

### 關鍵點：

- 此方法比 [build-image](#) 命令快。但是，這是一個手動過程，不能自動重複。

- 使用此方法，您無法存取可透過 CLI 取得的記錄擷取和映像生命週期管理命令。

步驟：

### New Amazon EC2 console

1. 查找與您使用的具體對應 AWS 區域的 AMI。要找到它，請使用帶有參數 `--region` 數的 `pcluster list-official-images` 命令來選擇特定的 AWS 區域和 `--os` 和 `--architecture` 參數，以便使用要使用的操作系統和架構過濾所需的 AMI。從輸出擷取 Amazon EC2 映像 ID。
2. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
3. 在導覽窗格中，選擇 [映像]，然後選擇 [AMI]。搜尋擷取的 EC2 映像 ID，選取 AMI，然後選擇從 AMI 啟動執行個體。
4. 向下捲動並選擇您的執行個體類型。
5. 選擇您的金鑰配對和啟動執行個體。
6. 使用作業系統使用者和 SSH 金鑰登入您的執行個體。
7. 手動自訂執行個體以符合您的需求。
8. 執行下列命令以準備 AMI 建立的執行個體。

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. 在主控台中，選擇執行個體狀態和停止執行個體。

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 Amazon EC2 主控台或建立映像檔，從執行個體建 AWS CLI [立](#)新的 AMI。

從 Amazon EC2 控制台

- 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，然後選擇建立映像。
- d. 選擇 Create Image (建立映像)。

11. 在叢集配置的 [CustomAmi](#) 欄位中輸入新的 AMI ID 並建立叢集。

## Old Amazon EC2 console

1. 查找與您使用的具體對應 AWS 區域的 AWS ParallelCluster AMI。要找到它，您可以使用帶有參數 `--region` 的 `pcluster list-official-images` 命令來選擇特定的 AWS 區域，`--os` 和 `--architecture` 參數，以使用要使用的操作系統和架構過濾所需的 AMI。您可以從輸出擷取 Amazon EC2 映像 ID。
2. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
3. 在導覽窗格中，選擇 [映像]，然後選擇 [AMI]。設定公用映像的篩選器，並搜尋擷取的 EC2 映像 ID，選取 AMI，然後選擇啟動。
4. 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執行個體。
5. 選擇 [啟動]，選取您的金鑰配對，然後啟動執行個體。
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，選取新執行個體並 Connect。
7. 手動自訂執行個體以符合您的需求。
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. 在 Amazon EC2 主控台中，選擇導覽窗格中的執行個體，選取新執行個體，然後選擇 [動作]、[執行個體狀態] 和 [停止]。
10. 使用 Amazon EC2 主控台或建立映像檔，從執行個體建 AWS CLI [立](#)新的 AMI。

從 Amazon EC2 控制台

- 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，然後選擇建立映像。
- d. 選擇 Create Image (建立映像)。

11. 在叢集配置的 [CustomAmi](#) 欄位中輸入新的 AMI ID 並建立叢集。

## 整合作用中目錄

在本自學課程中，您將建立多重使用者環境。此環境包括一 AWS ParallelCluster 個與 AWS Managed Microsoft AD (活動目錄) 在集成的 `corp.example.com`。您可以設定 Admin 使用者來管理目錄、讀

取目錄的ReadOnly使user000用者，以及登入叢集的使用者。您可以使用自動化路徑或手動路徑來建立聯網資源、作用中目錄 (AD) 和用於設定 AD 的 Amazon EC2 執行個體。無論路徑為何，您建立的基礎結構都會預先設定為 AWS ParallelCluster 使用下列其中一種方法進行整合：

- 具有證書驗證的 LDAPS ( 建議作為最安全的選項 )
- 無需驗證憑證的 LDAPS
- LDAP

LDAP 本身不提供加密。為了確保可能敏感資訊的安全傳輸，我們強烈建議您針對與 AD 整合的叢集使用 LDAPS (透過 TLS/SSL 的 LDAP)。如需詳細資訊，請參閱《AWS Directory Service 管理指南》AWS Managed Microsoft AD中的使用[啟用伺服器端 LDAPS](#)。

建立這些資源之後，請繼續設定並建立與您的 Active Directory (AD) 整合的叢集。建立叢集之後，請以您建立的使用者身分登入。若要取得有關在此自學課程中建立之規劃的更多資訊，請參閱[多個使用者存取叢集](#)〈[DirectoryService](#)規劃〉一節。

本教學課程介紹如何建立支援多個使用者存取叢集的環境。本教學課程不涵蓋如何建立和使用 AWS Directory Service AD。您 AWS Managed Microsoft AD 在本自學課程中設置所採取的步驟僅用於測試目的。它們不是為了取代官方文檔和最佳實踐，您可以在AWS Directory Service 管理指南[AWS Managed Microsoft AD](#)和 [Simple AD](#) 中找到。

#### Note

目錄使用者密碼會根據目錄密碼原則內容定義而過期。有關詳情，請參閱[支援的策略設定](#)。若要使用重設目錄密碼 AWS ParallelCluster，請參閱[如何重置用戶密碼和過期的密碼](#)。

#### Note

目錄網域控制站 IP 位址可能會因為網域控制站變更和目錄維護而變更。如果您選擇自動化的快速建立方法來建立目錄基礎結構，則必須在目錄 IP 位址變更時，手動對齊目錄控制器前面的負載平衡器。如果您使用快速建立方法，目錄 IP 位址不會自動與負載平衡器對齊。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

### 必要條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 `pcluster` CLI 所需[權限](#)的 IAM 角色。

當您瀏覽本教學課程時，請以您自己的名稱 `region-id` 和 ID 取代 *inputs highlighted in redd-abcdef01234567890*，例如和 `0123456789012` 用您的 AWS 帳戶 號碼替換。

### 步驟 1：建立 AD 基礎架構

選擇 [自動化] 索引標籤以建立具有 AWS CloudFormation 快速建立範本的 Active Directory (AD) 基礎結構。

選擇 [手動] 索引標籤以手動建立 AD 基礎結構。

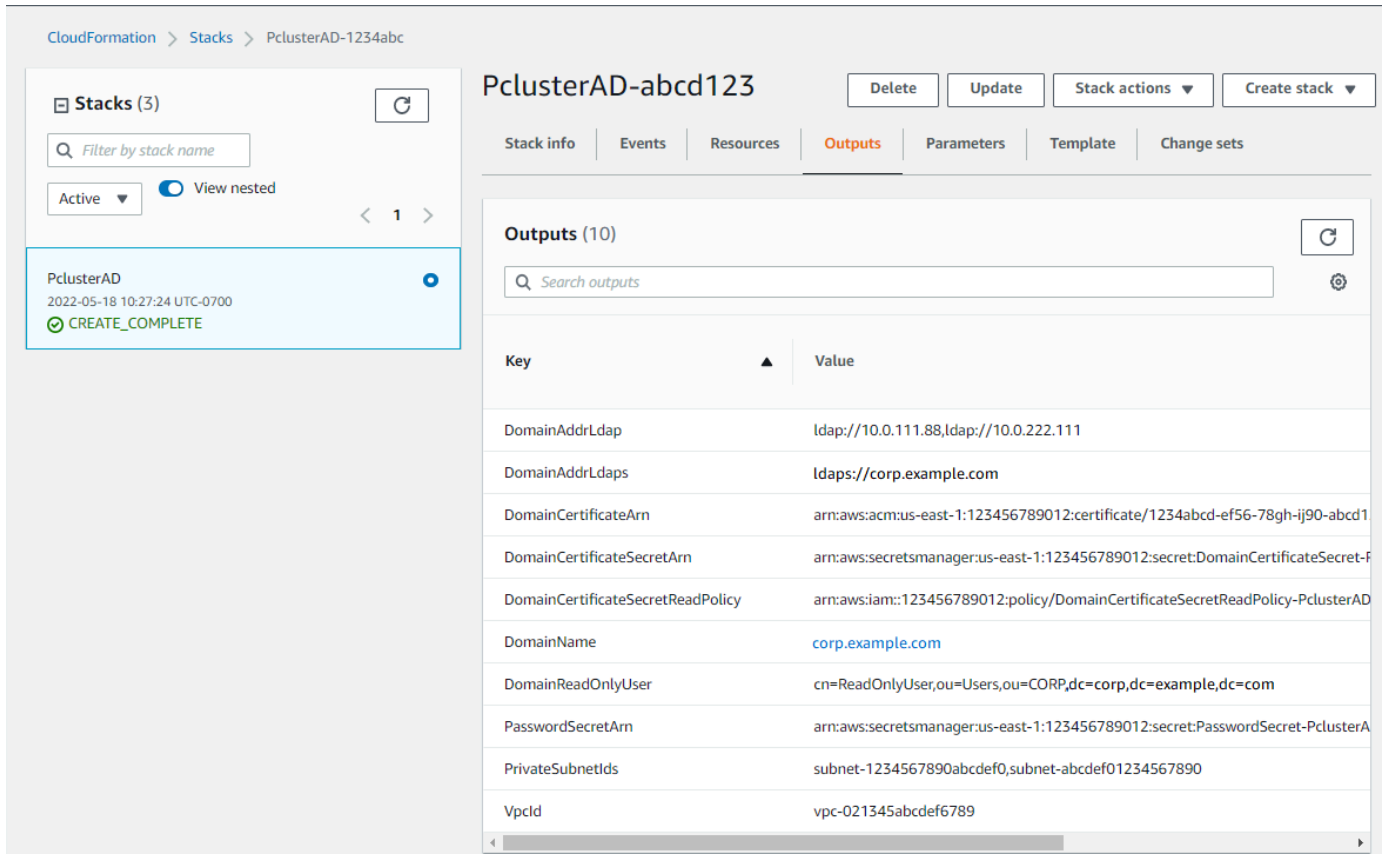
#### 自動化

1. 登入 AWS Management Console。
2. 開啟「[CloudFormation 快速建立](#)」(區域 `us-east-1`)，在主控台中建立下列資源：CloudFormation
  - 具有兩個子網路的 VPC 並路由供公用存取 (如果未指定 VPC)。
  - 一個 AWS Managed Microsoft AD。
  - 加入 AD 的 Amazon EC2 執行個體，您可以用來管理目錄。
3. 在快速建立堆疊頁面「參數」段落中，輸入下列參數的密碼：
  - AdminPassword
  - ReadOnlyPassword
  - UserPassword

記下密碼。您稍後在本自學課程中使用它們。

4. 針對 DomainName，請輸入 `corp.example.com`
5. 在 key pair 中，輸入 Amazon EC2 金鑰配對的名稱。

- 核取方塊以確認頁面底部的每個存取功能。
- 選擇建立堆疊。
- CloudFormation 堆疊到達CREATE\_COMPLETE狀態後，選擇堆疊的「輸出」索引標籤。記下輸出資源名稱和 ID，因為您需要在後面的步驟中使用它們。輸出提供建立叢集所需的資訊。



The screenshot shows the AWS CloudFormation console for a stack named 'PclusterAD-abcd123'. The stack is in the 'CREATE\_COMPLETE' state. The 'Outputs' tab is selected, displaying a table of 10 outputs. The table has columns for 'Key' and 'Value'.

| Key                               | Value                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------|
| DomainAddrLdap                    | ldap://10.0.111.88,ldap://10.0.222.111                                         |
| DomainAddrLdaps                   | ldaps://corp.example.com                                                       |
| DomainCertificateArn              | arn:aws:acm:us-east-1:123456789012:certificate/1234abcd-ef56-78gh-ij90-abcd1   |
| DomainCertificateSecretArn        | arn:aws:secretsmanager:us-east-1:123456789012:secret:DomainCertificateSecret-F |
| DomainCertificateSecretReadPolicy | arn:aws:iam::123456789012:policy/DomainCertificateSecretReadPolicy-PclusterAD  |
| DomainName                        | corp.example.com                                                               |
| DomainReadOnlyUser                | cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com                     |
| PasswordSecretArn                 | arn:aws:secretsmanager:us-east-1:123456789012:secret>PasswordSecret-PclusterA  |
| PrivateSubnetIds                  | subnet-1234567890abcdef0,subnet-abcdef01234567890                              |
| VpcId                             | vpc-021345abcdef6789                                                           |

- 若要完成練習(選擇性) [步驟 2：管理 AD 使用者和群組](#)，您需要目錄 ID。選擇資源並向下捲動以記下目錄 ID。
- 在(選擇性) [步驟 2：管理 AD 使用者和群組](#)或繼續[步驟 3：建立叢集](#)。

## 手動

為目錄服務建立 VPC，其中包含兩個子網路位於不同可用區域和 AWS Managed Microsoft AD

## 創建廣告

### Note

- 目錄和網域名稱為corp.example.com。簡短名稱是CORP。
- 變更指令Admin碼中的密碼。

- 活動目錄 (AD) 至少需要 15 分鐘才能創建。

使用下列 Python 指令碼在本機中建立 VPC、子網路 and AD 資源。AWS 區域將此文件另存為 `ad.py` 並運行它。

```
import boto3
import time
from pprint import pprint

vpc_name = "PclusterVPC"
ad_domain = "corp.example.com"
admin_password = "asdfASDF1234"

Amazon EC2 = boto3.client("ec2")
ds = boto3.client("ds")
region = boto3.Session().region_name

# Create the VPC, Subnets, IGW, Routes
vpc = ec2.create_vpc(CidrBlock="10.0.0.0/16")["Vpc"]
vpc_id = vpc["VpcId"]
time.sleep(30)
ec2.create_tags(Resources=[vpc_id], Tags=[{"Key": "Name", "Value": vpc_name}])
subnet1 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.0.0/17",
    AvailabilityZone=f"{region}a")["Subnet"]
subnet1_id = subnet1["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet1_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/
subnet1"}])
ec2.modify_subnet_attribute(SubnetId=subnet1_id, MapPublicIpOnLaunch={"Value": True})
subnet2 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.128.0/17",
    AvailabilityZone=f"{region}b")["Subnet"]
subnet2_id = subnet2["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet2_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/
subnet2"}])
ec2.modify_subnet_attribute(SubnetId=subnet2_id, MapPublicIpOnLaunch={"Value": True})
igw = ec2.create_internet_gateway()["InternetGateway"]
ec2.attach_internet_gateway(InternetGatewayId=igw["InternetGatewayId"], VpcId=vpc_id)
route_table = ec2.describe_route_tables(Filters=[{"Name": "vpc-id", "Values":
    [vpc_id]}])["RouteTables"][0]
ec2.create_route(RouteTableId=route_table["RouteTableId"],
    DestinationCidrBlock="0.0.0.0/0", GatewayId=igw["InternetGatewayId"])
```

```
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsSupport={"Value": True})
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsHostnames={"Value": True})

# Create the Active Directory
ad = ds.create_microsoft_ad(
    Name=ad_domain,
    Password=admin_password,
    Description="ParallelCluster AD",
    VpcSettings={"VpcId": vpc_id, "SubnetIds": [subnet1_id, subnet2_id]},
    Edition="Standard",
)
directory_id = ad["DirectoryId"]

# Wait for completion
print("Waiting for the directory to be created...")
directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
directory = directories[0]
while directory["Stage"] in {"Requested", "Creating"}:
    time.sleep(3)
    directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
    directory = directories[0]

dns_ip_addrs = directory["DnsIpAddrs"]

pprint({"directory_id": directory_id,
        "vpc_id": vpc_id,
        "subnet1_id": subnet1_id,
        "subnet2_id": subnet2_id,
        "dns_ip_addrs": dns_ip_addrs})
```

以下是來自 Python 腳本的示例輸出。

```
{
  "directory_id": "d-abcdef01234567890",
  "dns_ip_addrs": ["192.0.2.254", "203.0.113.237"],
  "subnet1_id": "subnet-021345abcdef6789",
  "subnet2_id": "subnet-1234567890abcdef0",
  "vpc_id": "vpc-021345abcdef6789"
}
```

記下輸出資源名稱和 ID。您可以在稍後的步驟中使用它們。



指令碼完成後，請繼續執行下一個步驟。

## 建立 Amazon EC2 執行個體

### New Amazon EC2 console

1. 登入 AWS Management Console。
2. 如果您沒有附加步驟 4 中列出的政策的角色，請在 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。否則，請跳至步驟 5。
3. 建立 ResetUserPassword 政策，將紅色反白顯示的內容取代為您的 AWS 區域 ID、帳戶 ID 和您執行建立 AD 之指令碼輸出的目錄 ID。

### ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. 建立已附加下列政策的 IAM 角色。
  - AWS [亞馬遜](#)管理策略 ManagedInstanceCore
  - AWS [亞馬遜](#)管理策略 DirectoryServiceAccess
  - ResetUserPassword 政策
5. 前往 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
6. 在 Amazon EC2 儀表板中，選擇啟動執行個體。
7. 在應用程式和作業系統映像中，選取最近使用的 Amazon Linux 2 AMI。
8. 在「執行個體類型」中，選擇「t2.micro」。
9. 在 key pair 中，選擇金鑰配對。
10. 針對網路設定，選擇編輯。

11. 對於 VPC，請選取目錄 VPC。
12. 向下滾動並選擇高級詳細信息。
13. 在 [進階詳細資料] 中，選擇 [corp.example.com網域加入目錄]
14. 對於 IAM 執行個體設定檔，請選擇您在步驟 1 中建立的角色，或選擇附加步驟 4 中列出政策的角色。
15. 在摘要中選擇啟動實例。
16. 記下執行個體識別碼 (例如 i-1234567890abcdef0)，然後等待執行個體完成啟動。
17. 執行個體啟動後，請繼續進行下一個步驟。

## Old Amazon EC2 console

1. 登入 AWS Management Console。
2. 如果您沒有附加步驟 4 中列出的政策的角色，請在 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。否則，請跳至步驟 5。
3. 建立ResetUserPassword策略。將紅色反白顯示的內容替換為您的 AWS 區域 AWS 帳戶 ID，ID 和目錄 ID，從您運行以創建活動目錄 (AD) 的腳本的輸出中。

### ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. 建立已附加下列政策的 IAM 角色。
  - AWS [亞馬遜](#)管理策略 ManagedInstanceCore
  - AWS [亞馬遜](#)管理策略 DirectoryServiceAccess
  - ResetUserPassword 政策

5. 前往 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
6. 在 Amazon EC2 儀表板中，選擇啟動執行個體。
7. 在應用程式和作業系統映像中，選取最近使用的 Amazon Linux 2 AMI。
8. 對於 Instance type (執行個體類型)，選擇 t2.micro。
9. 在 key pair 中，選擇金鑰配對。
10. 在網路設定中，選擇編輯。
11. 在 [網路設定] 的 [VPC] 中，選取目錄 VPC。
12. 向下滾動並選擇高級詳細信息。
13. 在 [進階詳細資料] 中，選擇 [**corp.example.com**網域加入目錄]
14. 在 [進階詳細資料] 的 [執行個體] 設定檔中，選擇您在步驟 1 中建立的角色，或選擇附加步驟 4 中所列原則的角色。
15. 在摘要中選擇啟動實例。
16. 記下執行個體 ID (例如i-1234567890abcdef0)，然後等待執行個體完成啟動。
17. 執行個體啟動後，請繼續進行下一個步驟。

## 將您的執行個體加入 AD

1. Connect 至您的執行個體，並以 AD 領域的身分加入 **admin**。

執行下列命令以連線至執行個體。

```
$ INSTANCE_ID="i-1234567890abcdef0"
```

```
$ PUBLIC_IP=$(aws ec2 describe-instances \  
--instance-ids $INSTANCE_ID \  
--query "Reservations[0].Instances[0].PublicIpAddress" \  
--output text)
```

```
$ ssh -i ~/.ssh/keys/keypair.pem ec2-user@$PUBLIC_IP
```

2. 安裝必要的軟件並加入領域。

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-  
common-tools krb5-workstation openldap-clients polycoreutils-python
```

### 3. 使用您的密碼取代管理員admin密碼。

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com  
Password for Admin:
```

如果上述項目已成功，表示您已加入範圍，並且可以繼續進行下一個步驟。

### 將使用者新增至 AD

#### 1. 建立 ReadOnlyUser 和其他使用者。

在此步驟中，您可以使用在上一個步驟中安裝的 [adcli](#) 和 [openldapm 用戶端](#) 工具。

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --  
display-name=ReadOnlyUser ReadOnlyUser
```

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --  
display-name=user000 user000
```

#### 2. 確認已建立使用者：

目錄 DNS IP 地址是 Python 腳本的輸出。

```
$ DIRECTORY_IP="192.0.2.254"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b  
"cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b  
"cn=user000,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

依預設，當您使用建立使用者時ad-cli，會停用該使用者。

#### 3. 從本機電腦重設並啟用使用者密碼：

登出您的亞馬遜 EC2 執行個體。

**Note**

- `ro-p@ssw0rd` 是密碼 `ReadOnlyUser`，從中擷取 AWS Secrets Manager。
- `user-p@ssw0rd` 是您 `connect (ssh)` 到叢集時所提供的叢集使用者密碼。

`directory-id` 這是 Python 指令碼的輸出。

```
$ DIRECTORY_ID="d-abcdef01234567890"
```

```
$ aws ds reset-user-password \  
--directory-id $DIRECTORY_ID \  
--user-name "ReadOnlyUser" \  
--new-password "ro-p@ssw0rd" \  
--region "region-id"
```

```
$ aws ds reset-user-password \  
--directory-id $DIRECTORY_ID \  
--user-name "user000" \  
--new-password "user-p@ssw0rd" \  
--region "region-id"
```

#### 4. 將密碼新增至密碼 Secrets Manager 密碼。

現在您已建立 `ReadOnlyUser` 並設定密碼，請將密碼儲存在 AWS ParallelCluster 用於驗證登入的密碼中。

使用 Secrets Manager 建立新密碼，以保留 `ReadOnlyUser` 作為值的密碼。密碼值格式必須僅為純文字 (非 JSON 格式)。記下秘密 ARN 以便 future 採取的步驟。

```
$ aws secretsmanager create-secret --name "ADSecretPassword" \  
--region region_id \  
--secret-string "ro-p@ssw0rd" \  
--query ARN \  
--output text  
arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
```

## 具有憑證驗證 (建議) 設定的 LDAPS

記下資源 ID。您可以在稍後的步驟中使用它們。

1. 在本機產生網域憑證。

```
$ PRIVATE_KEY="corp-example-com.key"
CERTIFICATE="corp-example-com.crt"
printf ".\n.\n.\n.\n.\n.\ncorp.example.com\n.\n.\n" | openssl req -x509 -sha256 -nodes -
newkey rsa:2048 -keyout $PRIVATE_KEY -days 365 -out $CERTIFICATE
```

2. 將憑證儲存至 Secrets Manager，以便稍後從叢集內擷取憑證。

```
$ aws secretsmanager create-secret --name example-cert \
  --secret-string file://$CERTIFICATE \
  --region region-id
{
  "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
  "Name": "example-cert",
  "VersionId": "14866070-092a-4d5a-bcdd-9219d0566b9c"
}
```

3. 將下列政策新增至您為將 Amazon EC2 執行個體加入 AD 網域而建立的 IAM 角色。

### PutDomainCertificateSecrets

```
{
  "Statement": [
    {
      "Action": [
        "secretsmanager:PutSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
      ],
      "Effect": "Allow"
    }
  ]
}
```

#### 4. 將憑證匯入至 AWS Certificate Manager (ACM)。

```
$ aws acm import-certificate --certificate fileb://$CERTIFICATE \  
--private-key fileb://$PRIVATE_KEY \  
--region region-id \  
{  
  "CertificateArn": "arn:aws:acm:region-  
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"  
}
```

#### 5. 建立和放在作用中目錄端點前面的負載平衡器。

```
$ aws elbv2 create-load-balancer --name CorpExampleCom-NLB \  
--type network \  
--scheme internal \  
--subnets subnet-1234567890abcdef0 subnet-021345abcdef6789 \  
--region region-id \  
{  
  "LoadBalancers": [  
    {  
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-  
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",  
      "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",  
      "CanonicalHostedZoneId": "Z2IF0LAFXWL04F",  
      "CreatedTime": "2022-05-05T12:56:55.988000+00:00",  
      "LoadBalancerName": "CorpExampleCom-NLB",  
      "Scheme": "internal",  
      "VpcId": "vpc-021345abcdef6789",  
      "State": {  
        "Code": "provisioning"  
      },  
      "Type": "network",  
      "AvailabilityZones": [  
        {  
          "ZoneName": "region-idb",  
          "SubnetId": "subnet-021345abcdef6789",  
          "LoadBalancerAddresses": []  
        },  
        {  
          "ZoneName": "region-ida",  
          "SubnetId": "subnet-1234567890abcdef0",  
          "LoadBalancerAddresses": []  
        }  
      ]  
    }  
  ]  
}
```

```

    ],
    "IpAddressType": "ipv4"
  }
]
}

```

## 6. 建立以作用中目錄端點為目標的目標群組。

```

$ aws elbv2 create-target-group --name CorpExampleCom-Targets --protocol TCP \
  --port 389 \
  --target-type ip \
  --vpc-id vpc-021345abcdef6789 \
  --region region-id
{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
      "TargetGroupName": "CorpExampleCom-Targets",
      "Protocol": "TCP",
      "Port": 389,
      "VpcId": "vpc-021345abcdef6789",
      "HealthCheckProtocol": "TCP",
      "HealthCheckPort": "traffic-port",
      "HealthCheckEnabled": true,
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 10,
      "HealthyThresholdCount": 3,
      "UnhealthyThresholdCount": 3,
      "TargetType": "ip",
      "IpAddressType": "ipv4"
    }
  ]
}

```

## 7. 將作用中目錄 (AD) 端點註冊到目標群組中。

```

$ aws elbv2 register-targets --target-group-arn
arn:aws:elasticloadbalancing:region-id:123456789012:targetgroup/CorpExampleCom-
Targets/44577c583b695e81 \
  --targets Id=192.0.2.254,Port=389 Id=203.0.113.237,Port=389 \
  --region region-id

```



## 8. 使用憑證建立 LB 接聽程式。

```
$ aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region-id:123456789012:loadbalancer/net/
CorpExampleCom-NLB/3afe296bf4ba80d4 \
  --protocol TLS \
  --port 636 \
  --default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 \
  --ssl-policy ELBSecurityPolicy-TLS-1-2-2017-01 \
  --certificates CertificateArn=arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 \
  --region region-id
"Listeners": [
  {
    "ListenerArn": "arn:aws:elasticloadbalancing:region-id:123456789012:listener/
net/CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b",
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
    "Port": 636,
    "Protocol": "TLS",
    "Certificates": [
      {
        "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
      }
    ],
    "SslPolicy": "ELBSecurityPolicy-TLS-1-2-2017-01",
    "DefaultActions": [
      {
        "Type": "forward",
        "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
        "ForwardConfig": {
          "TargetGroups": [
            {
              "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81"
            }
          ]
        }
      }
    ]
  }
]
```

```

    ]
  }
]
}

```

9. 建立託管區域，使網域在叢集 VPC 中可搜尋。

```

$ aws route53 create-hosted-zone --name corp.example.com \
  --vpc VPCRegion=region-id,VPCId=vpc-021345abcdef6789 \
  --caller-reference "ParallelCluster AD Tutorial"
{
  "Location": "https://route53.amazonaws.com/2013-04-01/hostedzone/
Z09020002B5MZQNXMSJUB",
  "HostedZone": {
    "Id": "/hostedzone/Z09020002B5MZQNXMSJUB",
    "Name": "corp.example.com.",
    "CallerReference": "ParallelCluster AD Tutorial",
    "Config": {
      "PrivateZone": true
    },
    "ResourceRecordSetCount": 2
  },
  "ChangeInfo": {
    "Id": "/change/C05533343BF3IKSORW1TQ",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T13:21:53.863000+00:00"
  },
  "VPC": {
    "VPCRegion": "region-id",
    "VPCId": "vpc-021345abcdef6789"
  }
}

```

10. 建立以下列內容命名 **recordset-change.json** 的檔案。 **HostedZoneId** 是負載平衡器的標準託管區域 ID。

```

{
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "corp.example.com",
        "Type": "A",

```

```

    "Region": "region-id",
    "SetIdentifier": "example-active-directory",
    "AliasTarget": {
      "HostedZoneId": "Z2IF0LAFXWL04F",
      "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
      "EvaluateTargetHealth": true
    }
  }
}
]
}

```

11. 這次使用託管區域 ID，將資料集變更提交至託管區域。

```

$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNMSJUB \
--change-batch file://recordset-change.json
{
  "ChangeInfo": {
    "Id": "/change/C0137926I56R3GC7XW2Y",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
  }
}

```

12. **policy.json**使用下列內容建立政策文件。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-abc123"
      ],
      "Effect": "Allow"
    }
  ]
}

```

13. 建立以下內容命名 **policy.json** 的政策文件。

```
$ aws iam create-policy --policy-name ReadCertExample \  
--policy-document file://policy.json  
{  
  "Policy": {  
    "PolicyName": "ReadCertExample",  
    "PolicyId": "ANPAUUXUVBC42VZSI4LDY",  
    "Arn": "arn:aws:iam::123456789012:policy/ReadCertExample-efg456",  
    "Path": "/",  
    "DefaultVersionId": "v1",  
    "AttachmentCount": 0,  
    "PermissionsBoundaryUsageCount": 0,  
    "IsAttachable": true,  
    "CreateDate": "2022-05-05T13:42:18+00:00",  
    "UpdateDate": "2022-05-05T13:42:18+00:00"  
  }  
}
```

14. 繼續執行 [\(選擇性\) 步驟 2：管理 AD 使用者和群組](#) 或中的步驟 [步驟 3：建立叢集](#)。

### (選擇性) 步驟 2：管理 AD 使用者和群組

在此步驟中，您可以從已加入作用中交付 (AD) 網域的 Amazon EC2 Amazon Linux 2 執行個體管理使用者和群組。

如果您遵循自動化路徑，請重新啟動並登入做為自動化操作一部分建立的 AD 加入執行個體。

如果您遵循手動路徑，請在上述步驟中重新啟動並登入您建立並加入 AD 的執行個體。

在這些步驟中，您可以使用安裝在執行個體中的 [adcli](#) 和 [openldap-client 用戶端](#) 工具做為上述步驟的一部分。

登入已加入 AD 網域的 Amazon EC2 執行個體

1. 在 Amazon EC2 主控台中，選取在先前步驟中建立的無標題 Amazon EC2 執行個體。執行個體狀態可能為 [已停止]。
2. 如果執行個體狀態為 [已停止]，請選擇 [執行個體狀態]，然後選擇
3. 通過狀態檢查之後，請選取執行個體，然後選擇 [Connect]，然後 SSH 在執行個體中選擇。

在登入已加入 AD 的 Amazon EC2 Linux 2 執行個體時管理使用者和群組

當您使用 `-U "Admin"` 選項執行 `adcli` 命令時，系統會提示您輸入 AD Admin 密碼。您可以將 AD Admin 密碼納入命令 `ldapsearch` 中。

1. 建立使用者。

```
$ adcli create-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

2. 設定使用者密碼。

```
$ aws --region "region-id" ds reset-user-password --directory-id "d-  
abcdef01234567890" --user-name "clusteruser" --new-password "new-p@ssw0rd"
```

3. 建立群組。

```
$ adcli create-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

4. 將使用者新增至群組。

```
$ adcli add-member "clusterteam" "clusteruser" --domain "corp.example.com" -U  
"Admin"
```

5. 描述使用者和群組。

描述所有使用者。

```
$ ldapsearch "(&(objectClass=user))" -x -h "192.0.2.254" -b  
"DC=corp,DC=example,DC=com" -D  
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

描述特定使用者。

```
$ ldapsearch "(&(objectClass=user)(cn=clusteruser))"  
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D  
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

使用名稱模式描述所有使用者。

```
$ ldapsearch "(&(objectClass=user)(cn=user*))" -x -h "192.0.2.254" -b
"DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

描述屬於特定群組的所有使用者。

```
$ ldapsearch "(&(objectClass=user)
(memberOf=CN=clusterteam,OU=Users,OU=CORP,DC=corp,DC=example,DC=com))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

描述所有群組

```
$ ldapsearch "objectClass=group" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com"
-D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

描述特定群組

```
$ ldapsearch "(&(objectClass=group)(cn=clusterteam))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

6. 從群組中移除使用者。

```
$ adcli remove-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

7. 刪除使用者。

```
$ adcli delete-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

8. 刪除群組。

```
$ adcli delete-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

## 步驟 3：建立叢集

如果您尚未結束 Amazon EC2 執行個體，請立即執行。

此環境設定為建立可以根據使用中目錄 (AD) 驗證使用者的叢集。

建立簡單的叢集配置，並提供與連線至 AD 相關的設定。如需詳細資訊，請參閱 [DirectoryService](#) 一節。

選擇下列其中一個叢集配置，並將其複製到名為 `ldaps_config.yaml`、`ldaps_nocert_config.yaml`、或的檔案中 `ldap_config.yaml`。

我們建議您選擇含有憑證驗證的 LDAPS 組態。如果您選擇此組態，您也必須將啟動程序指令碼複製到名為的檔案 `active-directory.head.post.sh`。此外，您必須將其存放在 Amazon S3 儲存貯體中，如組態檔中所示。

具有憑證驗證組態的 LDAPS (建議使用)

#### Note

必須變更下列元件。

- `KeyName`：您的一個 Amazon EC2 密鑰對。
- `SubnetId` / `SubnetIds`: CloudFormation 快速建立堆疊 (自動化教學課程) 或 `python` 指令碼 (手動教學課程) 輸出中提供的其中一個子網路 ID。
- `Region`：您建立 AD 基礎結構的區域。
- `DomainAddr`：此 IP 位址是您廣告服務的其中一個 DNS 位址。
- `PasswordSecretArn`：包含的密碼的 Amazon 資源名稱 (ARN) 的密碼。 `DomainReadOnlyUser`
- `BucketName`：包含啟動程序指令碼的值區名稱。
- `AdditionalPolicies/Policy`：讀取域認證政策 `ReadCertExample` 的 Amazon 資源名稱 (ARN)。
- `CustomActions/OnNodeConfigured/Args`：保存域認證政策的秘密的 Amazon 資源名稱 (ARN)。

為了獲得更好的安全狀態，我們建議使用 `HeadNodeSsh//AllowedIps` 配置來限制對頭節點的 SSH 訪問。

Region: *region-id*

```
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: keypair
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::123456789012:policy/ReadCertExample
    S3Access:
      - BucketName: my-bucket
        EnableWriteAccess: false
        KeyName: bootstrap/active-directory/active-directory.head.post.sh
    CustomActions:
      OnNodeConfigured:
        Script: s3://my-bucket/bootstrap/active-directory/active-directory.head.post.sh
        Args:
          - arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc
          - /opt/parallelcluster/shared/directory_service/domain-certificate.crt
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue0
        ComputeResources:
          - Name: queue0-t2-micro
            InstanceType: t2.micro
            MinCount: 1
            MaxCount: 10
        Networking:
          SubnetIds:
            - subnet-abcdef01234567890
  DirectoryService:
    DomainName: corp.example.com
    DomainAddr: ldaps://corp.example.com
    PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
    DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
    LdapTlsCaCert: /opt/parallelcluster/shared/directory_service/domain-certificate.crt
    LdapTlsReqCert: hard
```

## 引導腳本



建立啟動程序檔案之後，並在將其上傳到 S3 儲存貯體之前，請執行 `chmod +x active-directory.head.post.sh` 以授予 AWS ParallelCluster 執行權限。

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

CERTIFICATE_SECRET_ARN="$1"
CERTIFICATE_PATH="$2"

[[ -z $CERTIFICATE_SECRET_ARN ]] && echo "[ERROR] Missing CERTIFICATE_SECRET_ARN" &&
exit 1
[[ -z $CERTIFICATE_PATH ]] && echo "[ERROR] Missing CERTIFICATE_PATH" && exit 1

source /etc/parallelcluster/cfnconfig
REGION="${cfn_region:?}"

mkdir -p $(dirname $CERTIFICATE_PATH)
aws secretsmanager get-secret-value --region $REGION --secret-id
$CERTIFICATE_SECRET_ARN --query SecretString --output text > $CERTIFICATE_PATH
```

## 沒有憑證驗證組態的 LDAPS

### Note

必須變更下列元件。

- KeyName：您的一個 Amazon EC2 密鑰對。
- SubnetId / SubnetIds：CloudFormation 快速建立堆疊 (自動化教學課程) 或 python 指令碼 (手動教學課程) 輸出中的其中一個子網 ID。
- Region：您建立 AD 基礎結構的區域。
- DomainAddr：此 IP 位址是您廣告服務的其中一個 DNS 位址。
- PasswordSecretArn：包含的密碼的 Amazon 資源名稱 (ARN) 的密碼。DomainReadOnlyUser

為了獲得更好的安全狀態，我們建議使用 HeadNode /SSH/ AllowedIps 配置來限制對頭節點的 SSH 訪問。

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: keypair
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
      ComputeResources:
        - Name: queue0-t2-micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: corp.example.com
  DomainAddr: ldaps://corp.example.com
  PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsReqCert: never
```

## LDAP 組態設定

### Note

必須變更下列元件。

- KeyName : 您的一個 Amazon EC2 密鑰對。
- SubnetId / SubnetIds: CloudFormation 快速建立堆疊 (自動化教學課程) 或 python 指令碼 (手動教學課程) 輸出中提供的其中一個子網路 ID。
- Region : 您建立 AD 基礎結構的區域。
- DomainAddr : 此 IP 位址是您廣告服務的其中一個 DNS 位址。

- PasswordSecretArn : 包含的密碼的 Amazon 資源名稱 ( ARN ) 的密碼。DomainReadOnlyUser

為了獲得更好的安全狀態，我們建議使用 HeadNode /SSH/ AllowedIps 配置來限制對頭節點的 SSH 訪問。

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: keypair
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
      ComputeResources:
        - Name: queue0-t2-micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://192.0.2.254,ldap://203.0.113.237
  PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

使用下列命令建立您的叢集。

```
$ pcluster create-cluster --cluster-name "ad-cluster" --cluster-configuration "./ldaps_config.yaml"
```

```
{
  "cluster": {
    "clusterName": "pcluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
    "region": "region-id",
    "version": 3.7.0,
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

## 步驟 4：以使用者身分 Connect 至叢集

您可以使用下列指令判斷叢集的狀態。

```
$ pcluster describe-cluster -n ad-cluster --region "region-id" --query "clusterStatus"
```

輸出如下。

```
"CREATE_IN_PROGRESS" / "CREATE_COMPLETE"
```

當狀態達到時"CREATE\_COMPLETE"，請使用建立的使用者名稱和密碼登入。

```
$ HEAD_NODE_IP=$(pcluster describe-cluster -n "ad-cluster" --region "region-id" --query headNode.publicIpAddress | xargs echo)
```

```
$ ssh user000@$HEAD_NODE_IP
```

您可以在中提供為新使用者建立的SSH金鑰，而不需要密碼即可登入/home/user000@HEAD\_NODE\_IP/.ssh/id\_rsa。

如果命ssh令成功，表示您已成功連線至叢集，並經過驗證可使用 Active Directory (AD) 的使用者身分。

## 步驟 5：清除

1. 從本機電腦刪除叢集。

```
$ pcluster delete-cluster --cluster-name "ad-cluster" --region "region-id"
{
```

```
"cluster": {
  "clusterName": "ad-cluster",
  "cloudformationStackStatus": "DELETE_IN_PROGRESS",
  "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
  "region": "region-id",
  "version": "3.7.0",
  "clusterStatus": "DELETE_IN_PROGRESS"
}
```

## 2. 檢查要刪除的叢集進度。

```
$ pcluster describe-cluster --cluster-name "ad-cluster" --region "region-id" --
query "clusterStatus"
"DELETE_IN_PROGRESS"
```

成功刪除叢集之後，請繼續執行下一個步驟。

## 自動化

### 刪除活動目錄資源

1. 從 <https://console.aws.amazon.com/cloudformation/>.
2. 在導覽窗格中，選擇 Stacks (堆疊)。
3. 從堆疊清單中選擇 AD 堆疊 (例如 pcluster-ad)。
4. 選擇刪除。

## 手動

1. 刪除 Amazon EC2 實例。
  - a. 在 <https://console.aws.amazon.com/ec2/> 中，選擇導覽窗格中的 [執行個體]。
  - b. 從執行個體清單中，選擇您建立的執行個體，以將使用者新增至目錄。
  - c. 選擇實例狀態，然後選擇終止實例。
2. 刪除託管區域。
  - a. recordset-delete.json 使用以下內容創建一個。在此範例中，HostedZoneId 是負載平衡器的標準託管區域 ID。

```
{
  "Changes": [
    {
      "Action": "DELETE",
      "ResourceRecordSet": {
        "Name": "corp.example.com",
        "Type": "A",
        "Region": "region-id",
        "SetIdentifier": "pcluster-active-directory",
        "AliasTarget": {
          "HostedZoneId": "Z2IF0LAFXWL04F",
          "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}
```

- b. 使用託管區域 ID 將資料集變更提交至託管區域。

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXMSJUB \
  --change-batch file://recordset-delete.json
{
  "ChangeInfo": {
    "Id": "/change/C04853642A0TH2TJ5NLNI",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T14:25:51.046000+00:00"
  }
}
```

- c. 刪除託管區域。

```
$ aws route53 delete-hosted-zone --id Z09020002B5MZQNXMSJUB
{
  "ChangeInfo": {
    "Id": "/change/C0468051QFABTVHMDEG9",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T14:26:13.814000+00:00"
  }
}
```

```
}
```

3. 刪除 LB 監聽器。

```
$ aws elbv2 delete-listener \  
  --listener-arn arn:aws:elasticloadbalancing:region-id:123456789012:listener/net/  
  CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b --region region-id
```

4. 刪除目標群組。

```
$ aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:region-  
  id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 --  
  region region-id
```

5. 刪除負載平衡器。

```
$ aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:region-  
  id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4 --  
  region region-id
```

6. 刪除叢集用來從 Secrets Manager 讀取憑證的原則。

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/  
  ReadCertExample
```

7. 刪除包含網域憑證的密碼。

```
$ aws secretsmanager delete-secret \  
  --secret-id arn:aws:secretsmanager:region-id:123456789012:secret:example-  
  cert-123abc \  
  --region region-id  
{  
  "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",  
  "Name": "example-cert",  
  "DeletionDate": "2022-06-04T16:27:36.183000+02:00"  
}
```

8. 從 ACM 刪除憑證。

```
$ aws acm delete-certificate \  

```

```
--certificate-arn arn:aws:acm:region-id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 --region region-id
```

9. 刪除活動目錄 ( AD ) 資源。

a. 從 python 腳本的輸出中獲取以下資源 IDad.py :

- 廣告識別碼
- AD 子網路識別碼
- 廣告 VPC ID

b. 執行下列命令刪除目錄。

```
$ aws ds delete-directory --directory-id d-abcdef0123456789 --region region-id
{
  "DirectoryId": "d-abcdef0123456789"
}
```

c. 列出 VPC 中的安全群組。

```
$ aws ec2 describe-security-groups --filters '[{"Name":"vpc-id","Values":["vpc-07614ade95ebad1bc"]}]' --region region-id
```

d. 刪除自訂安全性群組。

```
$ aws ec2 delete-security-group --group-id sg-021345abcdef6789 --region region-id
```

e. 刪除子網路。

```
$ aws ec2 delete-subnet --subnet-id subnet-1234567890abcdef --region region-id
```

```
$ aws ec2 delete-subnet --subnet-id subnet-021345abcdef6789 --region region-id
```

f. 描述網際網路閘道。

```
$ aws ec2 describe-internet-gateways \
  --filters Name=attachment.vpc-id,Values=vpc-021345abcdef6789 \
  --region region-id
{
  "InternetGateways": [
```



```
    "Attachments": [  
      {  
        "State": "available",  
        "VpcId": "vpc-021345abcdef6789"  
      }  
    ],  
    "InternetGatewayId": "igw-1234567890abcdef",  
    "OwnerId": "123456789012",  
    "Tags": []  
  }  
]  
}
```

- g. 卸離網際網路閘道。

```
$ aws ec2 detach-internet-gateway \  
  --internet-gateway-id igw-1234567890abcdef \  
  --vpc-id vpc-021345abcdef6789 \  
  --region region-id
```

- h. 刪除網際網路閘道。

```
$ aws ec2 delete-internet-gateway \  
  --internet-gateway-id igw-1234567890abcdef \  
  --region region-id
```

- i. 刪除 VPC。

```
$ aws ec2 delete-vpc \  
  --vpc-id vpc-021345abcdef6789 \  
  --region region-id
```

- j. 刪除包含密ReadOnlyUser碼的密碼。

```
$ aws secretsmanager delete-secret \  
  --secret-id arn:aws:secretsmanager:region-  
id:123456789012:secret:ADSecretPassword-1234 \  
  --region region-id
```

# 使用 AWS KMS 金鑰設定共用儲存區加密

瞭解如何設定客戶受管 AWS KMS 金鑰，以便在設定的叢集檔案儲存系統中加密和保護您的資料 AWS ParallelCluster。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

AWS ParallelCluster 支援下列共用儲存裝置組態選項：

- [SharedStorage](#) / [EbsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [EfsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [FsxLustreSettings](#) / [KmsKeyId](#)

您可以使用這些選項為 Amazon EBS、Amazon EFS 和 FSx 提供用於 Lustre 共用儲存系統加密的客戶受管 AWS KMS 金鑰。若要使用它們，您必須針對下列項目建立和設定 IAM 政策：

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

## 必要條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 [pcluster](#) CLI 所需[權限](#)的 IAM 角色。

## 主題

- [建立 政策](#)
- [配置和建立叢集](#)

## 建立 政策

### 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. 選擇政策。
3. 選擇建立政策。
4. 選擇 JSON 索引標籤並貼上下列原則。確保將所有出現 `123456789012` 的 AWS 帳戶 ID 和密鑰 Amazon 資源名稱 (ARN) 替換為您自己 AWS 區域的 ID 和密鑰。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:region-id:123456789012:key/abcd1234-ef56-gh78-ij90-
        abcd1234efgh5678"
      ]
    }
  ]
}
```

5. 在本教學課程中，請為原則命名 `ParallelClusterKmsPolicy`，然後選擇 [建立原則]。
6. 記下原則 ARN。您需要它來配置您的叢集。

## 配置和建立叢集

以下是叢集組態範例，其中包含具有加密功能的 Amazon 彈性區塊存放區共用檔案系統。

```
Region: eu-west-1
```

```
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: my-ssh-key
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: q1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 0
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
SharedStorage:
  - MountDir: /shared/ebs1
    Name: shared-ebs1
    StorageType: Ebs
    EbsSettings:
      Encrypted: True
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

用您自己的值替換紅色文本中的項目。然後，建立一個叢集，使用您的 AWS KMS 金鑰加密 Amazon EBS 中的資料。

Amazon EFS 和 FSx 的組態類似於 Lustre 檔案系統。

Amazon EFS SharedStorage 組態如下。

```
...
SharedStorage:
```

```
- MountDir: /shared/efs1
  Name: shared-efs1
  StorageType: Efs
  EfsSettings:
    Encrypted: True
    KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

光澤SharedStorage組態的 FSx 如下所示。

```
...
SharedStorage:
  - MountDir: /shared/fsx1
    Name: shared-fsx1
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      DeploymentType: PERSISTENT_1
      PerUnitStorageThroughput: 200
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

## 在多佇列模式叢集中執行工作

本教學課程說明如何 AWS ParallelCluster 使用[多重佇列模式執行第一個 Hello World](#) 工作。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

### 必要條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 [pcluster](#) CLI 所需[權限](#)的 IAM 角色。

## 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 `pcluster version` 的相關資訊，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。

```
$ pcluster configure --config multi-queue-mode.yaml
```

如需 `pcluster configure` 命令的詳細資訊，請參閱 [pcluster configure](#)。

完成此步驟後，`multi-queue-mode.yaml` 會出現名為的基本組態檔案。此檔案包含基本叢集配置。

在下一個步驟中，您可以修改新的配置檔案，並啟動具有多個佇列的叢集。

#### Note

本教學課程使用的某些執行個體不符合免費方案資格。

在此自學課程中，請修改您的規劃檔以符合以下規劃。以紅色反白的項目代表您的組態檔案值。保持自己的價值觀。

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: c5.xlarge
Networking:
  SubnetId: subnet-abcdef01234567890
Ssh:
  KeyName: yourkeypair
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - Name: spot
    ComputeResources:
  - Name: c5xlarge
    InstanceType: c5.xlarge
    MinCount: 1
```

```
    MaxCount: 10
  - Name: t2micro
    InstanceType: t2.micro
    MinCount: 1
    MaxCount: 10
  Networking:
    SubnetIds:
      - subnet-abcdef01234567890
  - Name: ondemand
    ComputeResources:
      - Name: c52xlarge
        InstanceType: c5.2xlarge
        MinCount: 0
        MaxCount: 10
    Networking:
      SubnetIds:
        - subnet-021345abcdef6789
```

## 建立叢集

建立multi-queue-cluster根據您的組態檔案命名的叢集。

```
$ pcluster create-cluster --cluster-name multi-queue-cluster --cluster-configuration
multi-queue-mode.yaml
{
  "cluster": {
    "clusterName": "multi-queue-cluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

如需 pcluster create-cluster 命令的詳細資訊，請參閱[pcluster create-cluster](#)。

若要檢查叢集的狀態，請執行下列命令。

```
$ pcluster list-clusters
{
```

```
"cluster": {
  "clusterName": "multi-queue-cluster",
  "cloudformationStackStatus": "CREATE_IN_PROGRESS",
  "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
  "region": "eu-west-1",
  "version": "3.7.0",
  "clusterStatus": "CREATE_IN_PROGRESS"
}
```

建立叢集時，clusterStatus欄位會顯示CREATE\_COMPLETE。

## 登入頭節點

使用您的私密安全殼層金鑰檔案登入頭節點。

```
$ pcluster ssh --cluster-name multi-queue-cluster -i ~/path/to/yourkeyfile.pem
```

如需 pcluster ssh 的相關資訊，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
spot*      up    infinite   18   idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[1-9]
spot*      up    infinite    2   idle  spot-st-c5xlarge-1,spot-st-t2micro-1
ondemand   up    infinite   10   idle~ ondemand-dy-c52xlarge-[1-10]
```

輸出顯示您的叢集中有一個t2.micro和一個idle處於可用狀態的c5.xlarge運算節點。

其他節點全部處於省電狀態，由節點狀態的~尾碼表示，沒有 Amazon EC2 執行個體支援這些節點。預設佇列的佇列名稱後會以\*尾碼表示。spot是您的預設工作佇列。

## 以多重佇列模式執行工作

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。請確定目前使用者可以執行此指令碼。



```
$ tee <<EOF hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from \$(hostname)"
EOF

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

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 [sbatch](#) Slurm 文件中的。

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 1
```

您可以使用 `squeue` 指令檢視佇列並检查工作狀態。因為您未指定特定佇列，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)
   1      spot      wrap ec2-user  R       0:10     2 spot-st-c5xlarge-1,spot-st-
t2micro-1
```

輸出顯示任務目前處於執行中狀態。等待工作完成。這大約需要 30 秒。然後，`squeue` 再次運行。

```
$ squeue
JOBID PARTITION      NAME      USER  ST      TIME  NODES NODELIST(REASON)
```

現在佇列中的工作都已完成，請尋找在目前目錄 `slurm-1.out` 中命名的輸出檔案。

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```

輸出顯示工作在 `spot-st-t2micro-1` 和 `spot-st-c5xlarge-1` 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 2
```

您將這些參數用於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。這會要求一個c5.xlarge節點和兩個t2.micro節點用於此工作。

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
spot*      up    infinite   1     alloc# spot-dy-t2micro-1
spot*      up    infinite  17     idle~  spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*      up    infinite   1     mix    spot-st-c5xlarge-1
spot*      up    infinite   1     alloc  spot-st-t2micro-1
ondemand   up    infinite  10     idle~  ondemand-dy-c52xlarge-[1-10]
```

節點正在開啟電源。這由節點狀態的#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)
   2    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，節點即可使用，且工作會進入 R (RUNNING) 狀態。

```
$ squeue
JOBID PARTITION   NAME     USER ST       TIME  NODES NODELIST(REASON)
   2    spot      wrap ec2-user R        0:07     3 spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-t2micro-1
```

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
spot*      up    infinite   17   idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot*      up    infinite    3   idle  spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand   up    infinite   10   idle~ ondemand-dy-c52xlarge-[1-10]
```

然後，在佇列中沒有任何工作後，請 `slurm-2.out` 在本機目錄中檢查。

```
$ cat slurm-2.out
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
Hello World from spot-st-c5xlarge-1
```

這是叢集的最終狀態。

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
spot*      up    infinite   17   idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot*      up    infinite    3   idle  spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand   up    infinite   10   idle~ ondemand-dy-c52xlarge-[1-10]
```

登出叢集之後，您可以透過執行來進行清理 `pcluster delete-cluster`。如需更多詳細資訊，請參閱 [pcluster list-clusters](#) 及 [pcluster delete-cluster](#)。

```
$ pcluster list-clusters
{
  "clusters": [
    {
      "clusterName": "multi-queue-cluster",
      "cloudformationStackStatus": "CREATE_COMPLETE",
      "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
      "region": "eu-west-1",
      "version": "3.1.4",
      "clusterStatus": "CREATE_COMPLETE"
    }
  ]
}
$ pcluster delete-cluster -n multi-queue-cluster
{
```

```
"cluster": {
  "clusterName": "multi-queue-cluster",
  "cloudformationStackStatus": "DELETE_IN_PROGRESS",
  "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
  "region": "eu-west-1",
  "version": "3.1.4",
  "clusterStatus": "DELETE_IN_PROGRESS"
}
```

## 應用 AWS ParallelCluster 程式介面

在本教學中，您將使用 [Amazon API Gateway](#) 和 [AWS ParallelCluster CloudFormation 範本建置和測試 API](#)。然後，您可以使用上可用的示例客戶端 GitHub 來使用 API。如需如何使用 API 的詳細資訊，請參閱 [AWS ParallelCluster API](#)。

本教學課程摘自《面向[公共部門的 HPC 客戶](#)研討會》。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

### 必要條件

- 已在您的計算環境中[安裝](#)並設定。AWS CLI
- AWS ParallelCluster 安裝在虛擬環境中。如需詳細資訊，請參閱[AWS ParallelCluster 在虛擬環境中安裝](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 [pcluster](#) CLI 所需[權限](#)的 IAM 角色。

## 步驟 1：使用 Amazon API Gateway 構建 API

留在您的主使用者目錄中並啟動您的虛擬環境：

1. 安裝有用的 JSON 命令列處理器。

```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. 運行以下命令以獲取您的 AWS ParallelCluster 版本並將其分配給環境變量。

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" |tee -a ~/.bashrc
```

3. 建立環境變數並為其指派您的地區 ID。

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" |tee -a ~/.bashrc
```

4. 執行下列命令以部署 API。

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" |tee -a ~/.bashrc
```

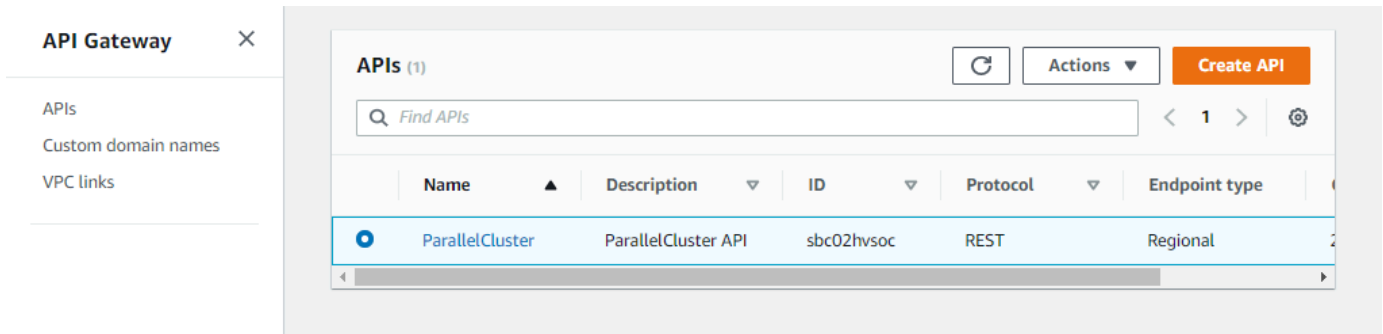
```
aws cloudformation create-stack \
  --region ${AWS_DEFAULT_REGION} \
  --stack-name ${API_STACK_NAME} \
  --template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
parallelcluster-api.yaml \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
  --parameters ParameterKey=EnableIamAdminAccess,ParameterValue=true

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-api-
stack/abcd1234-ef56-gh78-ei90-1234abcd5678"
}
```

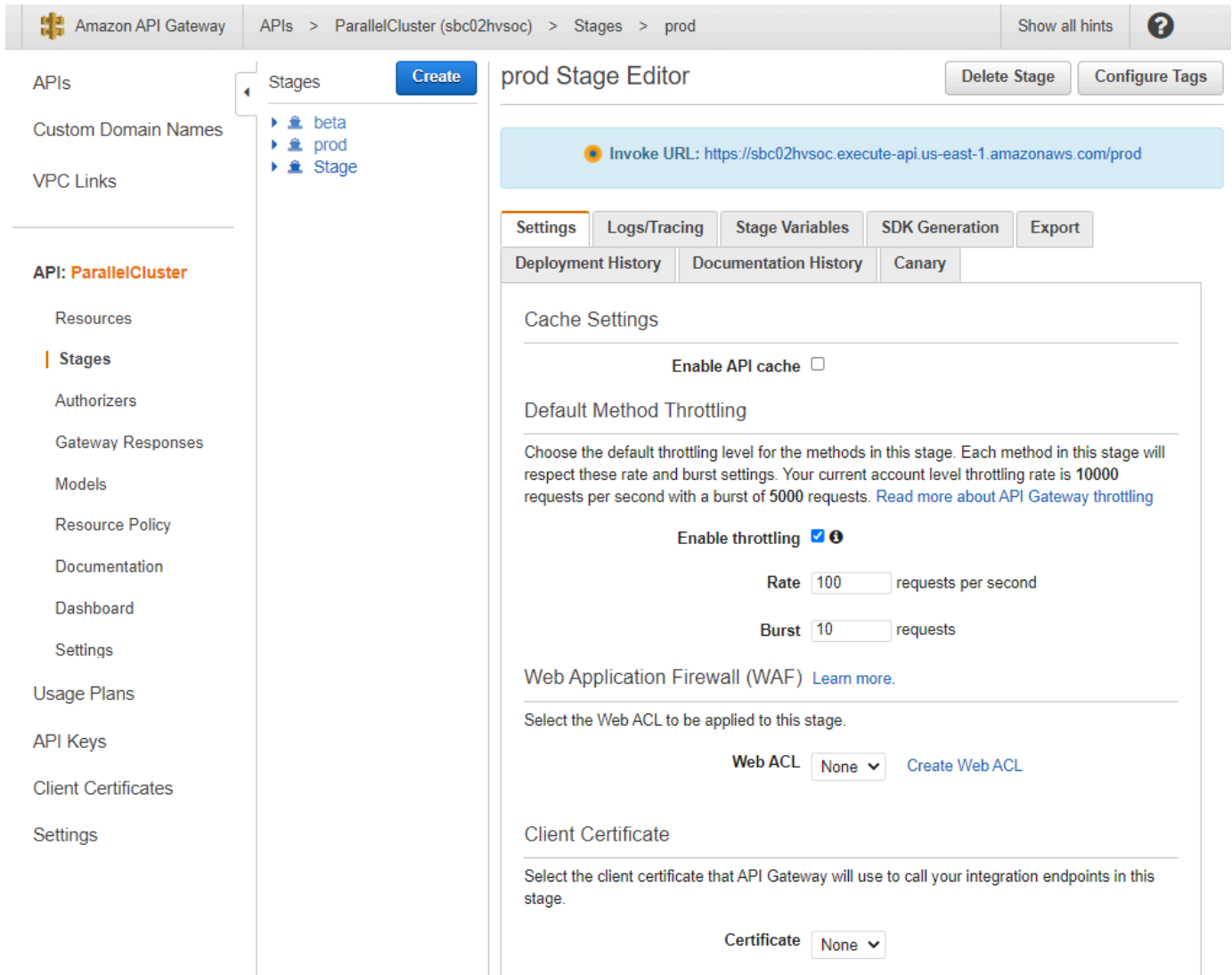
該過程完成後，繼續進行下一個步驟。

## 步驟 2：在 Amazon API Gateway 主控台中測試 API

1. 登入 AWS Management Console。
2. 瀏覽至 [Amazon API Gateway 主控台](#)。
3. 選擇您的 API 部署。



#### 4. 選擇階段並選取階段。



5. 請注意 API Gateway 為存取或叫用您的 API 所提供的 URL。它以藍色突出顯示。

6. 選擇資源，然後GET在下選取/**clusters**。

7. 選擇測試圖標，然後向下滾動並選擇測試圖標。

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET

Resources Actions **/v3/clusters - GET - Method Execution**

Method Request

- Auth: AWS IAM
- ARN: arn:aws:execute-api:us-east-1:123456789012:sbc02hvsoc\*/GET/v3/clusters
- Query Strings: region, nextToken, clusterStatus

Method Response

Select an integration response.

/clusters GET系統會顯示您的回應。

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfwk2) > GET

Show all hints ?

Resources Actions

Method Execution /v3/clusters - GET - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

**Path**  
No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.

**Request:** /v3/clusters  
**Status:** 200  
**Latency:** 3203 ms  
**Response Body**

**Query Strings**  
{clusters}  
param1=value1&param2=value2

**Headers**  
{clusters}  
Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

**Stage Variables**  
No stage variables exist for this method.

**Client Certificate**  
No client certificates have been generated.

```
{
  "clusters": [
    {
      "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/test-cluster/4450d850-b684-11ec-84a7-0a047567c9f3",
      "cloudformationStackStatus": "CREATE_COMPLETE",
      "clusterName": "test-cluster",
      "clusterStatus": "CREATE_COMPLETE",
      "region": "us-east-1",
      "version": "3.1.2"
    }
  ]
}
```

**Response Headers**

```
{"Content-Length": "360", "X-Amzn-Trace-Id": "Root=1-62686455-c1cf243417b2721e33822ac5;Sampled=1", "Content-Type": "application/json"}
```

Logs

### 步驟 3：準備並測試範例用戶端以叫用 API

將 AWS ParallelCluster 原始程式碼複製到 api 目錄中，然後安裝 Python 用戶端程式庫。

```
1. $ git clone -b v${PCLUSTER_VERSION} https://github.com/aws/aws-parallelcluster aws-parallelcluster-v${PCLUSTER_VERSION}
   cd aws-parallelcluster-v${PCLUSTER_VERSION}/api
```

```
$ pip3 install client/src
```

2. 導覽回您的主使用者目錄。



### 3. 匯出用戶端在執行時使用的 API Gateway 基礎 URL。

```
$ export PCLUSTER_API_URL=$( aws cloudformation describe-stacks
--stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiInvokeUrl`].OutputValue' --output text )
echo "export PCLUSTER_API_URL=${PCLUSTER_API_URL}" |tee -a ~/.bashrc
```

### 4. 匯出用戶端用來建立叢集的叢集名稱。

```
$ export CLUSTER_NAME="test-api-cluster"
echo "export CLUSTER_NAME=${CLUSTER_NAME}" |tee -a ~/.bashrc
```

### 5. 執行下列命令來儲存範例用戶端用來存取 API 的認證。

```
$ export PCLUSTER_API_USER_ROLE=$( aws cloudformation describe-
stacks --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiUserRole`].OutputValue' --output text )
echo "export PCLUSTER_API_USER_ROLE=${PCLUSTER_API_USER_ROLE}" |tee -a ~/.bashrc
```

## 步驟 4：複製用戶端程式碼指令碼並執行叢集測試

### 1. 將下列範例用戶端程式碼複製到您 `test_pcluster_client.py` 的 home 使用者目錄中。客戶端代碼發出請求以執行以下操作：

- 建立 叢集
- 描述叢集。
- 列出叢集。
- 描述運算叢集。
- 說明叢集執行個體。

```
# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
this
# software and associated documentation files (the "Software"), to deal in the
Software
```

```
# without restriction, including without limitation the rights to use, copy,
# modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and
# to
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
# COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)
    fleet_ops = cluster_compute_fleet_api.ClusterComputeFleetApi(api_client)
    instance_ops = cluster_instances_api.ClusterInstancesApi(api_client)
```

```
# Create cluster
build_done = False
try:
    with open('cluster-config.yaml', encoding="utf-8") as f:
        body = CreateClusterRequestContent(cluster_name=cluster_name,
cluster_configuration=f.read())
        api_response = cluster_ops.create_cluster(body, region=region)
except pcluster_client.ApiException as e:
    print("Exception when calling create_cluster: %s\n" % e)
    build_done = True
time.sleep(60)

# Confirm cluster status with describe_cluster
while not build_done:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('CREATE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
        elif api_response.cluster_status == ClusterStatus('CREATE_COMPLETE'):
            print('READY!')
            build_done = True
        else:
            print('ERROR!!!!')
            build_done = True
    except pcluster_client.ApiException as e:
        print("Exception when calling describe_cluster: %s\n" % e)

# List clusters
try:
    api_response = cluster_ops.list_clusters(region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling list_clusters: %s\n" % e)

# DescribeComputeFleet
try:
    api_response = fleet_ops.describe_compute_fleet(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling compute fleet: %s\n" % e)
```

```
# DescribeClusterInstances
try:
    api_response = instance_ops.describe_cluster_instances(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling describe_cluster_instances: %s\n" % e)
```

## 2. 建立叢集配置。

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

3. API 用戶端程式庫會自動偵測環境變數 (例如 `AWS_ACCESS_KEY_ID`、`AWS_SECRET_ACCESS_KEY`、或 `AWS_SESSION_TOKEN`) 或的組態詳細資料 `$HOME/.aws`。下列命令會將您目前的 IAM 角色切換至指定的角色 `ParallelClusterApiUserRole`。

```
$ eval $(aws sts assume-role --role-arn ${PCLUSTER_API_USER_ROLE} --role-session-name ApiTestSession | jq -r '.Credentials | "export AWS_ACCESS_KEY_ID=\(.AccessKeyId)\nexport AWS_SECRET_ACCESS_KEY=\(.SecretAccessKey)\nexport AWS_SESSION_TOKEN=\(.SessionToken)\n"')
```

要注意的錯誤：

如果您看到類似以下內容的錯誤，表示您已經假設 `ParallelClusterApiUserRole` 並且 `AWS_SESSION_TOKEN` 已過期。

```
An error occurred (AccessDenied) when calling the AssumeRole operation:
User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/ApiTestSession
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::XXXXXXXXXXXX:role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

卸除角色，然後重新執行 `aws sts assume-role` 命令以使用 `ParallelClusterApiUserRole`。

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

若要提供目前的使用者 API 存取權限，您必須[展開資源策略](#)。

4. 執行下列命令來測試範例用戶端。

```
$ python3 test_pcluster_client.py
{'cluster_configuration': 'Region: us-east-1\n'
                          'Image:\n'
                          '  Os: alinux2\n'
                          'HeadNode:\n'
                          '  InstanceType: t2.micro\n'
                          '  Networking . . . :\n'
                          '    SubnetId: subnet-1234567890abcdef0\n'
                          '  Ssh:\n'
                          '    KeyName: adpc\n'
                          'Scheduling:\n'
                          '  Scheduler: slurm\n'
                          '  SlurmQueues:\n'
                          '    - Name: queue1\n'
                          '      ComputeResources:\n'
                          '        - Name: t2micro\n'
                          '          InstanceType: t2.micro\n'
                          '          MinCount: 0\n'
                          '          MaxCount: 10\n'
                          '      Networking . . . :\n'
                          '        SubnetIds:\n'
                          '          - subnet-1234567890abcdef0\n',
  'cluster_name': 'test-api-cluster'}
{'cloud_formation_stack_status': 'CREATE_IN_PROGRESS',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
  'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-not-delete...'},
  'cluster_name': 'test-api-cluster',
  'cluster_status': 'CREATE_IN_PROGRESS',
  'compute_fleet_status': 'UNKNOWN',
  'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'region': 'us-east-1',
  'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
  'version': '3.1.3'}
```

```

.
.
. . . working . . . {'cloud_formation_stack_status': 'CREATE_COMPLETE',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
  'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
  'cluster_name': 'test-api-cluster',
  'cluster_status': 'CREATE_COMPLETE',
  'compute_fleet_status': 'RUNNING',
  'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
  'head_node': {'instance_id': 'i-abcdef01234567890',
    'instance_type': 't2.micro',
    'launch_time': datetime.datetime(2022, 4, 28, 16, 21, 46,
tzinfo=tzlocal()),
    'private_ip_address': '172.31.27.153',
    'public_ip_address': '52.90.156.51',
    'state': 'running'},
  'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
  'region': 'us-east-1',
  'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
  'version': '3.1.3'}
READY!

```

## 步驟 5：複製用戶端程式碼指令碼並刪除叢集

1. 將下列範例用戶端程式碼複製到 `delete_cluster_client.py`。用戶端程式碼會發出刪除叢集的要求。

```

# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
this
# software and associated documentation files (the "Software"), to deal in the
Software
# without restriction, including without limitation the rights to use, copy,
modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and
to

```

```
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
# COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)

    # Delete the cluster
    gone = False
    try:
        api_response = cluster_ops.delete_cluster(cluster_name, region=region)
    except pcluster_client.ApiException as e:
        print("Exception when calling delete_cluster: %s\n" % e)
```

```

time.sleep(60)

# Confirm cluster status with describe_cluster
while not gone:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('DELETE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
    except pcluster_client.ApiException as e:
        gone = True
        print("DELETE COMPLETE or Exception when calling describe_cluster: %s
\n" % e)

```

## 2. 執行下列命令以刪除叢集。

```

$ python3 delete_cluster_client.py
{'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 53, 48,
tzinfo=tzlocal()),
'private_ip_address': '172.31.17.132',
'public_ip_address': '34.201.100.37',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
.
.
.

```



```
. . . working . . . {'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
. . . working . . . DELETE COMPLETE or Exception when calling describe_cluster:
(404)
Reason: Not Found
.
.
.
HTTP response body: {"message": "Cluster 'test-api-cluster' does not exist or
belongs to an incompatible ParallelCluster major version."}
```

3. 完成測試之後，請取消設定環境變數。

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

## 步驟 6：清除

您可以使用 AWS Management Console 或 AWS CLI 刪除您的 API。

1. 從主 AWS CloudFormation 控制台選擇 API 堆疊，然後選擇 [刪除]。
2. 如果使用，請執行下列命令 AWS CLI。

使用 AWS CloudFormation.

```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```

## 使用Slurm會計建立叢集

瞭解如何使用Slurm帳戶設定和建立叢集。如需詳細資訊，請參閱 [Slurm會計與 AWS ParallelCluster](#)。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS 免費方案類別中使用它。如需詳細資訊，請參閱 [AWS ParallelClusterUI 成本](#)。

在本教學課程中，您會使用[CloudFormation 快速建立範本 \(us-east-1\)](#) 來建立 MySQL 無伺服器 [Amazon Aurora](#) 資料庫。範本會指示 CloudFormation 建立所有必要元件，以便在與叢集相同的 VPC 上部署 Amazon Aurora 無伺服器資料庫。此範本也會為叢集與資料庫之間的連線建立基本的網路和安全性組態。

### Note

從版本 3.3.0 開始，AWS ParallelCluster 支援使用叢集配置參數 [SlurmSettings/資料庫](#) 進行 Slurm 帳戶。

### Note

快速創建模板作為一個例子。此範本不涵蓋 Slurm 會計資料庫伺服器所有可能使用案例。您有責任建立具有適合生產工作負載之組態和容量的資料庫伺服器。

事前準備：

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 你有一個 [Amazon EC2 key pair](#)。
- 您擁有具有執行 [pcluster](#) CLI 所需 [權限](#) 的 IAM 角色。
- 您在中部署快速建立範本的區域支援 Amazon Aurora MySQL 無伺服器 v2。如需詳細資訊，請參閱 [使用 Aurora MySQL 的 Aurora 無伺服器 v2](#)。

## 步驟 1：建立下列項目的 VPC 和子網路 AWS ParallelCluster

若要使用為 Slurm 會計資料庫提供的 CloudFormation 範本，您必須準備好叢集的 VPC。您可以手動執行此操作，也可以作為 [使用 AWS ParallelCluster 命令行界面配置和創建集群](#) 程序的一部分進行。如果您已經使用過 AWS ParallelCluster，則可能已準備好用於部署叢集和資料庫伺服器的 VPC。

## 步驟 2：建立資料庫堆疊

使用 [CloudFormation 快速創建模板 \( us-east-1 \)](#) 創建用於會計的數據庫堆棧。Slurm 該模板需要以下輸入：

- 資料庫伺服器認證，特別是管理員使用者名稱和密碼。
- 調整 Amazon Aurora 無伺服器叢集的大小。這取決於預期的叢集載入。
- 網路參數，特別是用於建立子網路的目標 VPC 和子網路或 CIDR 區塊。

為您的資料庫伺服器選取適當的認證和大小。對於網路選項，您必須使用與 AWS ParallelCluster 叢集部署到的相同 VPC。您可以建立資料庫的子網路，並將它們當做輸入傳遞至範本。或者，為兩個子網路提供兩個脫離的 CIDR 區塊，並讓 CloudFormation 範本為 CIDR 區塊建立兩個子網路。請確定 CIDR 區塊不會與現有的子網路重疊。如果 CIDR 區塊與現有子網路重疊，則無法建立堆疊。

資料庫伺服器需要幾分鐘的時間來建立。

## 步驟 3：建立啟用 Slurm 會計功能的叢集

提供的 CloudFormation 模板生成具有一些定義輸出的 CloudFormation 堆棧。從中 AWS Management Console，您可以在 CloudFormation 堆疊檢視的 [輸出] 索引標籤中檢視輸出。若要啟用 Slurm 帳戶，必須在 AWS ParallelCluster 叢集配置檔案中使用其中一些輸出：

- DatabaseHost：用於 [SlurmSettings/Database/Uri](#) 集群配置參數。
- DatabaseAdminUser：用於 [SlurmSettings/Database/UserName](#) 叢集配置參數值。
- DatabaseSecretArn：用於 [SlurmSettings/Database/PasswordSecretArn](#) 集群配置參數。
- DatabaseClientSecurityGroup：這是附加到集群的頭節點，這是在 [HeadNode/Networking/SecurityGroups](#) 配置參數中定義的安全組。

使用輸出值更新叢集配置檔案 Database 參數。使用 [pcluster](#) CLI 建立叢集。

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

建立叢集之後，您可以開始使用Slurm帳號指令，例如sacctmgr或sacct。

## 使用外部Slurmdbd帳戶建立叢集

了解如何使用外部帳戶設定和建立叢集。Slurmdbd如需詳細資訊，請參閱[使用的Slurm帳戶](#) AWS ParallelCluster。

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[使用的AWS 服務 AWS ParallelCluster](#)。

AWS ParallelCluster UI 建立在無伺服器架構上，在大多數情況下，您可以在 AWS Free Tier類別中使用它。如需詳細資訊，請參閱 [AWS ParallelCluster UI 成本](#)。

在本教學課程中，您會使用 AWS CloudFormation 快速建立範本來建立必要的元件，以便在與叢集相同的 VPC 上部署 Slurmdbd 執行個體。此範本會為叢集與資料庫之間的連線建立基本的網路和安全性設定。

### Note

從開始version 3.10.0，使用群集設定參數 AWS ParallelCluster 支援外部 Slurmdbd。SlurmSettings / ExternalSlurmdbd

### Note

快速創建模板作為一個例子。此模板不涵蓋所有可能的用例。您有責任建立具有適合您生產工作負載的設定和容量的外部 Slurmdbd。

事前準備：

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有一個 [Amazon 彈性計算雲 key pair](#)。
- 您具有執行 [pcluster](#) CLI 所需[權限](#)的 AWS Identity and Access Management 角色。
- 你有一個Slurm會計數據庫。若要逐步完成建立Slurm會計資料庫的教學課程，請遵循[建立 Slurm 會計資料庫](#)堆疊中的步驟 1 和 2。

## 第 1 步：創建思路堆棧

在本教程中，使用 [CloudFormation 快速創建模板 \( us-east-1 \)](#) 創建 Slurmdbd 堆棧。該模板需要以下輸入：

### 聯網

- VPCid：用來啟動實例的虛擬私人雲端識別碼。
- SubnetId：用來啟動 Slurmdbd 執行個體的子網路識別碼。
- PrivatePrefix：虛擬私人 VPC 的 CIDR 前置詞。
- PrivateIp：要指派給 Slurmdbd 執行個體的次要私有 IP。

### 資料庫連線

- 安全性群組：要附加至 Slurmdbd 執行個體的安全性群組。此安全群組應允許資料庫伺服器和 Slurmdbd 執行個體之間的連線。
- 數據庫管理系統 DatabaseName：數據庫的名稱。
- DBM 使用者名稱：資料庫的使用者名稱。
- DBMS PasswordSecretArn：包含密碼到數據庫的秘密。
- DBSURI：資料庫伺服器的 URI。

### 執行個體設定

- InstanceType：要用於 slurmdbd 執行個體的執行個體類型。
- KeyName：用於 slurmdbd 執行個體的 Amazon EC2 key pair。

### 藍牙設置

- AMIID：思魯姆德實例的 AMI。AMI 應該是 ParallelCluster AMI。ParallelCluster AMI 的版本確定思魯姆德的版本。
- MungeKeySecretArn：包含用於驗證 Slurmdbd 和叢集之間通訊的 munge 金鑰的秘密。
- SlurmdbdPort：所使用的連接埠號碼。
- EnableSlurmdbdSystemService：啟用 slurmdbd 作為系統服務，並在執行個體啟動時執行。

**⚠ Warning**

如果資料庫是由不同版本的建立SlurmDB，請勿做Slurmdbd為系統服務使用。  
如果資料庫包含大量項目，則Slurm Database Daemon (SlurmDBD)可能需要數十分鐘才能更新資料庫，而且在此時間間隔內沒有回應。  
升級之前SlurmDB，請先備份資料庫。如需詳細資訊，請參閱 [Slurm 文件](#)。

## 步驟 2：建立Slurmdbd啟用外部的叢集

提供的 AWS CloudFormation 模板生成一個包含一些已解決輸出的 AWS CloudFormation 堆棧。

從中檢視 AWS CloudFormation 堆疊中的 [輸出] 索引標籤 AWS Management Console，以檢閱建立的實體。要啟用會Slurm計，必須在 AWS ParallelCluster 配置中使用其中一些輸出：

- SlurmdbdPrivatelp：用於 [SlurmSettings/ExternalSlurmdbd/主機叢集設定參數](#)。
- SlurmdbdPort：用於 [SlurmSettings/ExternalSlurmdbd/連接埠叢集設定參數值](#)。
- AccountingClientSecurityGroup：這是附加到集群的頭節點的安全組，該組在 [HeadNode/網絡/AdditionalSecurityGroups](#) 設置參數中解除。

另外，從 AWS CloudFormation 堆棧視圖中的「參數」選項卡中：

- MungeKeySecretArn：用於 [SlurmSettings/MungeKeySecretArn](#) 叢集設定參數值。

使用輸出值更新您的叢集設定資料庫參數。使用 pcluster AWS CLI 來建立叢集。

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

建立叢集之後，您可以開始使用Slurm帳號指令，例如sacctmgr或sacct。

**⚠ Warning**

與外部ParallelCluster之間的流SlurmDB量未加密。建議您在受信任的網路SlurmDB中執行叢集和外部。

## 還原至先前的AWS系統管理員文件版本

瞭解如何還原至先前的AWS系統管理員文件版本。如需 SSM 文件的詳細資訊，請參閱[AWS系統管理員使用指南中的AWS系統管理員文件](#)。

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 使用的服務 AWS ParallelCluster](#)。

AWS ParallelClusterUI 建立在無伺服器架構上，在大多數情況下，您可以在AWS免費方案類別中使用它。如需詳細資訊，請參閱[AWS ParallelClusterUI 成本](#)。

事前準備：

- AWS 帳戶具有管理 SSM 文件的權限。
- AWS CLI已安裝並設定。

## 還原為先前的 SSM 文件版本

1. 在終端機中，執行下列命令以取得您擁有的現有 SSM 文件清單。

```
$ aws ssm list-documents --document-filter "key=Owner,value=Self"
```

2. 將 SSM 文件還原為先前的版本。在這個例子中，我們恢復到SessionManagerRunShell文檔的先前版本。您可以使用 SSM SessionManagerRunShell 文件來自訂您起始的每個 SSM 殼層工作階段。
  - a. 執行下列命令尋找的DocumentVersion參數：SessionManagerRunShell

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {
```

```

        "Name": "linuxcmd",
        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then
source /opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/
bash"
    }
],
"PlatformTypes": [
    "Windows",
    "Linux",
    "MacOS"
],
"DocumentType": "Session",
"SchemaVersion": "1.0",
"LatestVersion": "2",
"DefaultVersion": "1",
"DocumentFormat": "JSON",
"Tags": []
}
}

```

最新版本為 2。

- b. 執行下列命令以回復至先前的版本：

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-  
version 2
```

3. 再次執行describe-document指令，確認文件版本是否已回復：

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {

```



```
        "Name": "linuxcmd",
        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then source /
opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/bash"
    }
],
"PlatformTypes": [
    "Windows",
    "Linux",
    "MacOS"
],
"DocumentType": "Session",
"SchemaVersion": "1.0",
"LatestVersion": "1",
"DefaultVersion": "1",
"DocumentFormat": "JSON",
"Tags": []
}
}
```

最新版本為 1。

## 建立叢集 AWS CloudFormation

瞭解如何使用 AWS ParallelCluster CloudFormation 自訂資源建立叢集。如需詳細資訊，請參閱 [AWS CloudFormation 自訂資源](#)。

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時所建立的 AWS 資源付費。如需詳細資訊，請參閱 [AWS 使用的服務 AWS ParallelCluster](#)。

事前準備：

- AWS CLI [已安裝並設定](#)。
- 一個 [Amazon EC2 key pair](#)。
- 具有執行 [pcluster](#) CLI 所需[權限](#)的 IAM 角色。

## 使用 CloudFormation 快速建立堆疊建立叢集

在本教學課程中，您會使用快速建立堆疊來部署建立叢集和下列 AWS 資源的 CloudFormation 範本：

- 使用 CloudFormation 快速建立 CloudFormation 堆疊所建立的根堆疊。
- 包含預設原則、預設 VPC 設定和自訂資源提供者的巢狀 CloudFormation 堆疊。
- 範例 AWS ParallelCluster 叢集堆疊和叢集，您可以登入並執行作業。

## 建立叢集 AWS CloudFormation

1. 登入 AWS Management Console。
2. 開啟 CloudFormation [快速建立連結](#)，在 CloudFormation 主控台中建立下列資源：
  - 具有公用和私有子網路的 VPC 的巢狀 CloudFormation 堆疊，用於分別執行叢集頭節點和計算節點。
  - 具有用於管理叢集的 AWS ParallelCluster 自訂資源的巢狀 CloudFormation 堆疊。
  - 具有管理叢集之預設原則的巢狀 CloudFormation 堆疊。
  - 巢狀 CloudFormation 堆疊的根堆疊。
  - 具有Slurm排程器和已定義計算節點數目的 AWS ParallelCluster 叢集。

CloudFormation > Stacks > Create stack

## Quick create stack

### Template

Template URL  
https://pcluster-cfn-us-east-2.s3.amazonaws.com/parallelcluster/3.5.0/templates/custom\_resource/cluster-1-click.yaml

Stack description  
AWS ParallelCluster CloudFormation Cluster

### Stack name

Stack name  
cluster-0

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

### Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AvailabilityZone  
Availability zone where instances will be launched  
us-east-2a

KeyName  
KeyPair to login to the head node  
Select AWS::EC2::KeyPair::KeyName

### Capabilities

**The following resource(s) require capabilities: [AWS::CloudFormation::Stack]**  
This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: {0}. Check the capabilities of these resources. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY\_AUTO\_EXPAND

Cancel Create change set Create stack

3. 在「快速建立堆疊參數」區段中，輸入下列參數的值：
  - a. 在中 KeyName，輸入您的 Amazon EC2 key pair 的名稱。
  - b. 對於 AvailabilityZone，選擇叢集節點的 AZ，例如，us-east-1a。

- 核取方塊以確認頁面底部的每個存取功能。
- 選擇建立堆疊。
- 等待 CloudFormation 堆棧到達狀CREATE\_COMPLETE態。

## 使用 AWS CloudFormation 命令列介面 (CLI) 建立叢集

在本教學課程中，您會使用 AWS 命令列介面 (CLI) CloudFormation 來部署建立叢集的 CloudFormation 範本。

建立下列 AWS 資源：

- 使用 CloudFormation 快速建立 CloudFormation 堆疊所建立的根堆疊。
- 包含預設原則、預設 VPC 設定和自訂資源提供者的巢狀 CloudFormation 堆疊。
- 範例 AWS ParallelCluster 叢集堆疊和叢集，您可以登入並執行作業。

以您自己#####，例如###。

建立叢集 AWS CloudFormation

- 建立具有下列內cluster\_template.yaml容命名的 CloudFormation 範本：

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
  KeyName:
    Description: KeyPair to login to the head node
    Type: AWS::EC2::KeyPair::KeyName

  AvailabilityZone:
    Description: Availability zone where instances will be launched
    Type: AWS::EC2::AvailabilityZone::Name
    Default: us-east-2a

Mappings:
  ParallelCluster:
    Constants:
      Version: 3.7.0

Resources:
```

```
PclusterClusterProvider:
  Type: AWS::CloudFormation::Stack
  Properties:
    TemplateURL: !Sub
      - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
        ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
      - { Version: !FindInMap [ParallelCluster, Constants, Version] }

PclusterVpc:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      PublicCIDR: 10.0.0.0/24
      PrivateCIDR: 10.0.16.0/20
      AvailabilityZone: !Ref AvailabilityZone
    TemplateURL: !Sub
      - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
        ${AWS::URLSuffix}/parallelcluster/${Version}/templates/networking/public-private-
        ${Version}.cfn.json
      - { Version: !FindInMap [ParallelCluster, Constants, Version] }

PclusterCluster:
  Type: Custom::PclusterCluster
  Properties:
    ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
    ClusterName: !Sub 'c-${AWS::StackName}'
    ClusterConfiguration:
      Image:
        Os: alinux2
      HeadNode:
        InstanceType: t2.medium
        Networking:
          SubnetId: !GetAtt [ PclusterVpc , Outputs.PublicSubnetId ]
      Ssh:
        KeyName: !Ref KeyName
      Scheduling:
        Scheduler: slurm
        SlurmQueues:
          - Name: queue0
            ComputeResources:
              - Name: queue0-cr0
                InstanceType: t2.micro
            Networking:
              SubnetIds:
```

```

- !GetAtt [ PclusterVpc , Outputs.PrivateSubnetId ]

Outputs:
  HeadNodeIp:
    Description: The Public IP address of the HeadNode
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]

```

- 執行下列 AWS CLI 命令以部署 CloudFormation 堆疊以進行叢集建立和管理。

```

$ aws cloudformation deploy --template-file ./cluster_template.yaml \
  --stack-name mycluster \
  --parameter-overrides KeyName=keypair \
    AvailabilityZone=us-east-2b \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND

```

## 檢視 CloudFormation 叢集輸出

檢視 CloudFormation 叢集輸出以取得有用的叢集詳細資訊。新增的ValidationMessages屬性可讓您存取叢集建立和更新作業的驗證訊息。

- 導覽至主 [CloudFormation 控制台](#)，然後選取包含您 AWS ParallelCluster 自訂資源的堆疊。
- 選擇堆疊詳細資料，然後選取輸出索引標籤。

| Key                | Value                                                                                                                                                                                                                   | Description                                            |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| HeadNodeIp         | 1.2.3.4                                                                                                                                                                                                                 | The Public IP address of the HeadNode                  |
| ValidationMessages | [[{"level": "WARNING", "type": "KeyPairValidator", "message": "If you do not specify a key pair, you can't connect to the instance unless you choose an AMI that is configured to allow users another way to log in"}]] | Any warnings from cluster create or update operations. |

驗證訊息可能會被截斷。如需如何擷取記錄檔的詳細資訊，請參閱 [AWS ParallelCluster 疑難排](#)。

## 存取您的叢集

存取叢集。

### ssh進入叢集頭節點

- CloudFormation 堆疊部署完成後，請使用下列命令取得頭節點的 IP 位址：

```
$ HEAD_NODE_IP=$(aws cloudformation describe-stacks --stack-name=mycluster --query "Stacks|[0].Outputs[?OutputKey=='HeadNodeIp']|[0].OutputValue" --output=text)
```

您也可以從 CloudFormation 主控台的叢集堆疊輸出索引標籤中的HeadNodeIp參數擷取頭節點 IP 位址。

您可以在這裡找到頭節點 IP 位址，因為它是在叢集範 CloudFormation 本的Outputs區段中新增的，特別是針對此範例叢集。

2. 執行下列命令，Connect 至叢集頭節點：

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

## 清除

刪除叢集。

1. 執行下列 AWS CLI 命令以刪除 CloudFormation 堆疊和叢集。

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. 通過運行以下命令檢查堆棧刪除狀態。

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

## 使用地形部署 ParallelCluster API

在本教程中，您將定義一個簡單的 Terraform 項目來部署 API。ParallelCluster

先決條件

- 已安裝地形 v1.5.7 +。
- 具有部署 ParallelCluster API 權限的 IAM 角色。請參閱[the section called “所需的許可”](#)。

## 定義地形專案

1. 創建一個名為的目錄my-pcluster-api。

您建立的所有檔案都會位於此目錄中。

2. 建立檔案`provider.tf`以設定提 AWS 供者。

```
provider "aws" {
  region = var.region
  profile = var.profile
}
```

3. 使用 `ParallelCluster` 模組建立檔案`main.tf`以定義資源。

```
module "parallelcluster_pcluster_api" {
  source = "aws-tf/parallelcluster/aws//modules/pcluster_api"
  version = "1.0.0"

  region          = var.region
  api_stack_name  = var.api_stack_name
  api_version     = var.api_version
  deploy_pcluster_api = true
  parameters = {
    EnableIamAdminAccess = "true"
  }
}
```

4. 創建文件`variables.tf`以定義可以為此項目注入的變量。

```
variable "region" {
  description = "The region the ParallelCluster API is deployed in."
  type        = string
  default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

variable "api_stack_name" {
  type        = string
  description = "The name of the CloudFormation stack used to deploy the ParallelCluster API."
  default     = "ParallelCluster"
}
```



```
}  
  
variable "api_version" {  
  type      = string  
  description = "The version of the ParallelCluster API."  
}  
}
```

5. 建立檔案`terraform.tfvars`以設定變數的任意值。

下面的文件`us-east-1`使用堆棧名稱部署了一個 ParallelCluster API

3.10.0。MyParallelClusterAPI-310您將能夠使用其堆棧名稱引用此 ParallelCluster API 部署。

```
region = "us-east-1"  
api_stack_name = "MyParallelClusterAPI-310"  
api_version = "3.10.0"
```

6. 建立檔案`outputs.tf`以定義此專案傳回的輸出。

```
output "pcluster_api_stack_outputs" {  
  value = module.parallelcluster_pcluster_api.stack_outputs  
}
```

項目目錄是：

```
my-pcluster-api  
### main.tf - Terraform entrypoint to define the resources using the  
ParallelCluster module.  
### outputs.tf - Defines the outputs returned by Terraform.  
### providers.tf - Configures the AWS provider.  
### terraform.tfvars - Set the arbitrary values for the variables, i.e. region,  
PCAPI version, PCAPI stack name  
### variables.tf - Defines the variables, e.g. region, PCAPI version, PCAPI stack  
name.
```

## 部署 API

若要部署 API，請依序執行標準的地形命令。

1. 建立專案：

```
terraform init
```

## 2. 定義部署計劃：

```
terraform plan -out tfplan
```

## 3. 部署計劃：

```
terraform apply tfplan
```

## 所需的許可

您需要以下權限才能使用 Terraform 部署 ParallelCluster API：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:GetTemplate"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormationRead"
    },
    {
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/MyParallelClusterAPI*",
      "Effect": "Allow",
      "Sid": "CloudFormationWrite"
    },
    {
      "Action": [
        "cloudformation:CreateChangeSet"
      ],

```

```

    "Resource": [
      "arn:PARTITION:cloudformation:REGION:aws:transform/Include",
      "arn:PARTITION:cloudformation:REGION:aws:transform/
Serverless-2016-10-31"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormationTransformWrite"
  },
  {
    "Action": [
      "s3:GetObject"
    ],
    "Resource": [
      "arn:PARTITION:s3::*-aws-parallelcluster/parallelcluster/*/api/
ParallelCluster.openapi.yaml",
      "arn:PARTITION:s3::*-aws-parallelcluster/parallelcluster/*/layers/aws-
parallelcluster/lambda-layer.zip"
    ],
    "Effect": "Allow",
    "Sid": "S3ParallelClusterArtifacts"
  },
  {
    "Action": [
      "iam:CreateRole",
      "iam>DeleteRole",
      "iam:GetRole",
      "iam>CreatePolicy",
      "iam>DeletePolicy",
      "iam:GetPolicy",
      "iam:GetRolePolicy",
      "iam:AttachRolePolicy",
      "iam:DetachRolePolicy",
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy",
      "iam>ListPolicyVersions"
    ],
    "Resource": [
      "arn:PARTITION:iam::ACCOUNT:role/*",
      "arn:PARTITION:iam::ACCOUNT:policy/*"
    ],
    "Effect": "Allow",
    "Sid": "IAM"
  },
  {

```

```

    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:PARTITION:iam::ACCOUNT:role/ParallelClusterLambdaRole-*",
      "arn:PARTITION:iam::ACCOUNT:role/APIGatewayExecutionRole-*"
    ],
    "Effect": "Allow",
    "Sid": "IAMPassRole"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda:DeleteFunction",
      "lambda:GetFunction",
      "lambda:PublishLayerVersion",
      "lambda:DeleteLayerVersion",
      "lambda:GetLayerVersion",
      "lambda:TagResource",
      "lambda:UntagResource"
    ],
    "Resource": [
      "arn:PARTITION:lambda:REGION:ACCOUNT:layer:PCLayer-*",
      "arn:PARTITION:lambda:REGION:ACCOUNT:function:*-
ParallelClusterFunction-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
  },
  {
    "Action": [
      "logs:CreateLogGroup",
      "logs>DeleteLogGroup",
      "logs:DescribeLogGroups",
      "logs:PutRetentionPolicy",
      "logs:TagLogGroup",
      "logs:UntagLogGroup"
    ],
    "Resource": [
      "arn:PARTITION:logs:REGION:ACCOUNT:log-group:/aws/lambda/*-
ParallelClusterFunction-*"
    ],
    "Effect": "Allow",
    "Sid": "Logs"
  }
}

```

```
    },
    {
      "Action": [
        "apigateway:DELETE",
        "apigateway:GET",
        "apigateway:PATCH",
        "apigateway:POST",
        "apigateway:PUT",
        "apigateway:UpdateRestApiPolicy"
      ],
      "Resource": [
        "arn:PARTITION:apigateway:REGION:./restapis",
        "arn:PARTITION:apigateway:REGION:./restapis/*",
        "arn:PARTITION:apigateway:REGION:./tags/*"
      ],
      "Effect": "Allow",
      "Sid": "APIGateway"
    }
  ]
}
```

## 使用地形建立叢集

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時所建立的 AWS 資源付費。如需詳細資訊，請參閱 [the section called “AWS 使用的服務 AWS ParallelCluster”](#)。

### 先決條件

- 已安裝地形 v1.5.7 +。
- [the section called “AWS ParallelCluster API”](#)v3.8.0+ 已部署在您的帳戶中。請參閱[the section called “使用地形部署 ParallelCluster API”](#)。
- 具有呼叫 ParallelCluster API 之許可的 IAM 角色。請參閱 [所需權限]

## 定義地形專案

在本教學課程中，您將定義一個簡單的 Terraform 專案來部署叢集。

1. 創建一個名為的目錄my-clusters。

您建立的所有檔案都會位於此目錄中。

## 2. 建立 terraform.tf 要匯入 ParallelCluster 提供者的檔案。

```
terraform {
  required_version = ">= 1.5.7"
  required_providers {
    aws-parallelcluster = {
      source = "aws-tf/aws-parallelcluster"
      version = "1.0.0"
    }
  }
}
```

## 3. 建立檔案 providers.tf 以設定 ParallelCluster 和 AWS 提供者。

```
provider "aws" {
  region = var.region
  profile = var.profile
}

provider "aws-parallelcluster" {
  region          = var.region
  profile         = var.profile
  api_stack_name = var.api_stack_name
  use_user_role  = true
}
```

## 4. 使用 ParallelCluster 模組建立檔案 main.tf 以定義資源。

```
module "pcluster" {
  source = "aws-tf/parallelcluster/aws"
  version = "1.0.0"

  region          = var.region
  api_stack_name = var.api_stack_name
  api_version     = var.api_version
  deploy_pcluster_api = false

  template_vars = local.config_vars
  cluster_configs = local.cluster_configs
  config_path = "config/clusters.yaml"
}
```

## 5. 建立檔案 clusters.tf 以將多個叢集定義為 Terraform 局部變數。

**Note**

您可以在`cluster_config`元素中定義多個叢集。對於每個叢集，您可以在區域變數中明確定義叢集內容 (請參閱`DemoCluster01`) 或參考外部檔案 (請參閱`DemoCluster02`)。

若要檢閱可在組態元素中設定的叢集內容，請參閱[the section called “叢集配置檔”](#)。

若要檢閱可針對叢集建立設定的選項，請參閱[the section called “pcluster create-cluster”](#)。

```
locals {
  cluster_configs = {
    DemoCluster01 : {
      region : local.config_vars.region
      rollbackOnFailure : false
      validationFailureLevel : "WARNING"
      suppressValidators : [
        "type:KeyPairValidator"
      ]
      configuration : {
        Region : local.config_vars.region
        Image : {
          Os : "alinux2"
        }
        HeadNode : {
          InstanceType : "t3.small"
          Networking : {
            SubnetId : local.config_vars.subnet
          }
          Iam : {
            AdditionalIamPolicies : [
              { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
            ]
          }
        }
      }
      Scheduling : {
        Scheduler : "slurm"
        SlurmQueues : [{
          Name : "queue1"
          CapacityType : "ONDEMAND"
        }]
      }
    }
  }
}
```

```

    Networking : {
      SubnetIds : [local.config_vars.subnet]
    }
    Iam : {
      AdditionalIamPolicies : [
        { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
      ]
    }
    ComputeResources : [{
      Name : "compute"
      InstanceType : "t3.small"
      MinCount : "1"
      MaxCount : "4"
    }]
  }]
  SlurmSettings : {
    QueueUpdateStrategy : "TERMINATE"
  }
}
}
}
DemoCluster02 : {
  configuration : "config/cluster_config.yaml"
}
}
}
}

```

## 6. 建立檔案config/clusters.yaml以將多個叢集定義為 YAML 組態。

```

DemoCluster03:
  region: ${region}
  rollbackOnFailure: true
  validationFailureLevel: WARNING
  suppressValidators:
    - type:KeyPairValidator
  configuration: config/cluster_config.yaml
DemoCluster04:
  region: ${region}
  rollbackOnFailure: false
  configuration: config/cluster_config.yaml

```

## 7. 創建文件config/cluster\_config.yaml，這是一個標準的 ParallelCluster 配置文件，可以在其中注入 Terraform 變量。



若要檢閱可在組態元素中設定的叢集內容，請參閱[the section called “叢集配置檔”](#)。

```
Region: ${region}
Image:
  Os: alinux2
HeadNode:
  InstanceType: t3.small
  Networking:
    SubnetId: ${subnet}
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CapacityType: ONDEMAND
      Networking:
        SubnetIds:
          - ${subnet}
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
  ComputeResources:
    - Name: compute
      InstanceType: t3.small
      MinCount: 1
      MaxCount: 5
  SlurmSettings:
    QueueUpdateStrategy: TERMINATE
```

## 8. 建立檔案clusters\_vars.tf以定義可以插入叢集配置的變數。

此檔案可讓您定義可用於叢集配置的動態值，例如地區和子網路。

此範例會直接從專案變數擷取值，但您可能需要使用自訂邏輯來判斷這些值。

```
locals {
  config_vars = {
    subnet = var.subnet_id
    region = var.cluster_region
  }
}
```

```
}
```

## 9. 創建文件variables.tf以定義可以為此項目注入的變量。

```
variable "region" {
  description = "The region the ParallelCluster API is deployed in."
  type        = string
  default     = "us-east-1"
}

variable "cluster_region" {
  description = "The region the clusters will be deployed in."
  type        = string
  default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

variable "subnet_id" {
  type        = string
  description = "The id of the subnet to be used for the ParallelCluster
instances."
}

variable "api_stack_name" {
  type        = string
  description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
  default     = "ParallelCluster"
}

variable "api_version" {
  type        = string
  description = "The version of the ParallelCluster API."
}
```

## 10. 建立檔案terraform.tfvars以設定變數的任意值。

下列檔案會使用現有 ParallelCluster API 3.10.0 (已使用堆疊名稱部署在中) 在us-east-1子網路eu-west-1subnet-123456789內部署叢集。MyParallelClusterAPI-310

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

cluster_region = "eu-west-1"
subnet_id = "subnet-123456789"
```

11. 建立檔案outputs.tf以定義此專案傳回的輸出。

```
output "clusters" {
  value = module.pcluster.clusters
}
```

項目目錄是：

```
my-clusters
### config
#   ### cluster_config.yaml - Cluster configuration, where terraform variables can
#   be injected..
#   ### clusters.yaml - File listing all the clusters to deploy.
### clusters.tf - Clusters defined as Terraform local variables.
### clusters_vars.tf - Variables that can be injected into cluster configurations.
### main.tf - Terraform entrypoint where the ParallelCluster module is configured.
### outputs.tf - Defines the cluster as a Terraform output.
### providers.tf - Configures the providers: ParallelCluster and AWS.
### terraform.tf - Import the ParallelCluster provider.
### terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
### variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

## 部署叢集

若要部署叢集，請依序執行標準的 Terraform 指令。

### Note

此範例假設您已在帳戶中部署 ParallelCluster API。

## 1. 建立專案：

```
terraform init
```

## 2. 定義部署計劃：

```
terraform plan -out tfplan
```

## 3. 部署計劃：

```
terraform apply tfplan
```

## 使用叢集部署 ParallelCluster API

如果您尚未部署 ParallelCluster API，並且想要將其與叢集一起部署，請變更下列檔案：

- main.tf

```
module "pcluster" {
  source = "aws-tf/aws/parallelcluster"
  version = "1.0.0"

  region          = var.region
  api_stack_name  = var.api_stack_name
  api_version     = var.api_version
  deploy_pcluster_api = true

  template_vars      = local.config_vars
  cluster_configs    = local.cluster_configs
  config_path        = "config/clusters.yaml"
}
```

- providers.tf

```
provider "aws-parallelcluster" {
  region  = var.region
  profile = var.profile
  endpoint = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiInvokeUrl
  role_arn = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiUserRole
}
```

## 所需的許可

您需要下列權限才能使用 Terraform 部署叢集：

- 假設 ParallelCluster API 角色，負責與 ParallelCluster API 進行交互
- 描述 ParallelCluster API 的 AWS CloudFormation 堆棧以驗證它是否存在並檢索其參數和輸出

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:PARTITION:iam::ACCOUNT:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}
```

## 使用地形創建自定義 AMI

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時所建立的 AWS 資源付費。如需詳細資訊，請參閱 [the section called “AWS 使用的服務 AWS ParallelCluster”](#)。

先決條件

- 已安裝地形 v1.5.7 +。
- [the section called “AWS ParallelCluster API”](#) v3.8.0+ 已部署在您的帳戶中。請參閱 [the section called “使用地形建立叢集”](#)。
- 具有呼叫 ParallelCluster API 之許可的 IAM 角色。請參閱 [the section called “所需的許可”](#)。

## 定義地形專案

在本教程中，您將定義一個簡單的 Terraform 項目來部署 ParallelCluster 自定義 AMI。

1. 創建一個名為的目錄my-amis。

您建立的所有檔案都會位於此目錄中。

2. 建立terraform.tf要匯入 ParallelCluster 提供者的檔案。

```
terraform {
  required_version = ">= 1.5.7"
  required_providers {
    aws-parallelcluster = {
      source = "aws-tf/aws-parallelcluster"
      version = "1.0.0"
    }
  }
}
```

3. 建立檔案providers.tf以設定 ParallelCluster 和 AWS 提供者。

```
provider "aws" {
  region = var.region
  profile = var.profile
}

provider "aws-parallelcluster" {
  region          = var.region
  profile         = var.profile
  api_stack_name = var.api_stack_name
  use_user_role  = true
}
```

4. 使用 ParallelCluster模組建立檔案main.tf以定義資源。

若要檢閱可在image\_configuration元素中設定的影像性質，請參閱[the section called “構建映像配置文件”](#)。

若要檢閱可以為建立影像設定的選項，例如image\_id和rollback\_on\_failure，請參閱[the section called “pcluster build-image”](#)。

```
data "aws-parallelcluster_list_official_images" "parent_image" {
```

```
    region = var.region
    os = var.os
    architecture = var.architecture
}

resource "aws-parallelcluster_image" "demo01" {
  image_id          = "demo01"
  image_configuration = yamlencode({
    "Build":{
      "InstanceType": "c5.2xlarge",
      "ParentImage": data.aws-
parallelcluster_list_official_images.parent_image.official_images[0].amiId,
      "UpdateOsPackages": {"Enabled": false}
    }
  })
  rollback_on_failure = false
}
```

## 5. 創建文件variables.tf以定義可以為此項目注入的變量。

```
variable "region" {
  description = "The region the ParallelCluster API is deployed in."
  type        = string
  default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

variable "api_stack_name" {
  type        = string
  description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
  default     = "ParallelCluster"
}

variable "api_version" {
  type        = string
  description = "The version of the ParallelCluster API."
}
```

```
variable "os" {
  type      = string
  description = "The OS of the ParallelCluster image."
}

variable "architecture" {
  type      = string
  description = "The architecture of the ParallelCluster image."
}
```

## 6. 創建文件`terraform.tfvars`以為變量設置任意值。

透過以下檔案，可使用現有 ParallelCluster API 3.10.0 (已使用堆疊名稱部署在中) 部署自訂 AMI (適用us-east-1於 x86\_64) 架構的 Amazon Linux 2。us-east-1 MyParallelClusterAPI-310

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

os = "alinux2"
architecture = "x86_64"
```

## 7. 建立檔案`outputs.tf`以定義此專案傳回的輸出。

```
output "parent_image" {
  value = data.aws-
parallelcluster_list_official_images.parent_image.official_images[0]
}

output "custom_image" {
  value = aws-parallelcluster_image.demo01
}
```

項目目錄是：

```
my-amis
### main.tf - Terraform entrypoint where the ParallelCluster module is configured.
### outputs.tf - Defines the cluster as a Terraform output.
### providers.tf - Configures the providers: ParallelCluster and AWS.
### terraform.tf - Import the ParallelCluster provider.
### terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
```



```
### variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

## 部署 AMI

若要部署 AMI，請依序執行標準地形命令。

### 1. 建立專案：

```
terraform init
```

### 2. 定義部署計劃：

```
terraform plan -out tfplan
```

### 3. 部署計劃：

```
terraform apply tfplan
```

## 所需的許可

您需要下列權限才能使用 Terraform 部署自訂 AMI：

- 假設 ParallelCluster API 角色，負責與 ParallelCluster API 進行交互
- 描述 ParallelCluster API 的 AWS CloudFormation 堆棧，以驗證它是否存在並檢索其參數和輸出

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:PARTITION:iam::ACCOUNT:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],

```

```
    "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
    "Effect": "Allow",
    "Sid": "CloudFormation"
  }
]
```

## AWS ParallelCluster與身分識別中心的 UI 整合

本教學課程的目標是示範如何將 AWS ParallelCluster UI 與 IAM 身分中心整合，以提供單一登入解決方案，以統一 Active Directory 中可與AWS ParallelCluster叢集共用的使用者。

使用時AWS ParallelCluster，您只需為建立或更新AWS ParallelCluster映像和叢集時所建立的AWS資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

事前準備：

- 可以按照[此處的說明安裝現有的 AWS ParallelCluster UI](#)。
- 現有的受管理的活動目錄，最好是您也將用於[整合的](#)目錄AWS ParallelCluster。

## 啟用 IAM Identity Center

如果您已經將身分中心連接到您的AWS Managed Microsoft AD ( Active Directory )，則可以使用該身分中心，您可以跳到將應用程序添加到 IAM 身份中心部分。

如果您尚未將身分識別中心連線到AWS Managed Microsoft AD，請按照以下步驟進行設定。

啟用識別中心

1. 在主控台中，導覽至 IAM 身分中心。(請確保您在您擁有的地區AWS Managed Microsoft AD。)
2. 按一下「啟用」按鈕，這可能會詢問您是否要啟用組織，這是必要條件，因此您可以選擇啟用它。注意：這將通過電子郵件向您的帳戶管理員發送一封確認電子郵件，您應該按照鏈接進行確認。

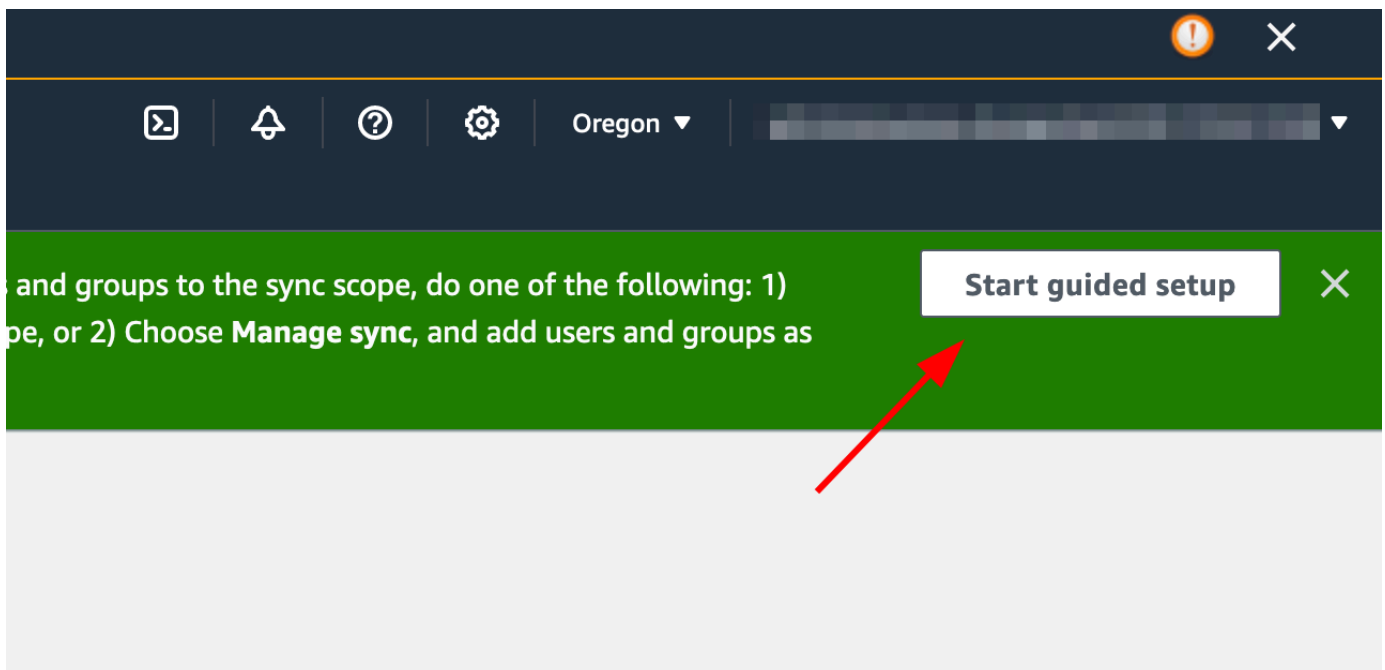
將身分識別中心連線至受管理

1. 在啟用身分識別中心後的下一頁上，您應該會看到建議的設定步驟，在步驟 1 下，選取 [選擇您的身分識別來源]。

2. 在 [身分識別來源] 區段中，按一下 [動作] 下拉式功能表 (位於右上角)，然後選取 [變更身分識別來源]。
3. 選取作用中目錄。
4. 在「現有目錄」下，選擇您的目錄。
5. 按一下 Next (下一步)。
6. 檢閱您的變更，捲動至底部，在文字方塊中輸入 ACCEPT 以確認，然後按一下「變更身分識別來源」。
7. 等待更改完成，然後您應該在頂部看到一個綠色橫幅。

### 同步使用者和群組至身分識別中心

1. 在綠色橫幅中，按一下 [開始引導式設定] (右上角的按鈕)



2. 在 [設定屬性對應] 中，按 [下一步]
3. 在 [設定同步範圍] 區段中，輸入要同步至身分識別中心的使用者名稱，然後按一下 [新增]
4. 新增使用者和群組後，按一下 [下一步]

**Users** | **Groups**

---

**User**

corp.pcluster.com ▼    🔍 Enter exact match user to add to sync scope    **Add**

---

**Added users and groups (4)** **Remove**

| <input type="checkbox"/> | Username / Group name | Type | Domain            |
|--------------------------|-----------------------|------|-------------------|
| <input type="checkbox"/> | user1                 | User | corp.pcluster.com |
| <input type="checkbox"/> | user2                 | User | corp.pcluster.com |
| <input type="checkbox"/> | admin1                | User | corp.pcluster.com |
| <input type="checkbox"/> | admin2                | User | corp.pcluster.com |

**Cancel**    **Previous**    **Next**

5. 檢閱您的變更，然後按一下儲存設定
6. 如果您在下一個畫面中看到有關使用者未同步的警告，請選取右上角的 [繼續同步] 按鈕。
7. 接下來，若要啟用使用者，請在左側的 [使用者] 索引標籤中選取使用者，然後按一下 [啟用使用者存取權] > [啟用使用者]

注意：如果頂端有警告橫幅，您可能需要選取 [繼續同步]，然後等待使用者進行同步處理 (請嘗試重新整理按鈕，查看他們是否已同步處理)。

The screenshot displays the IAM Identity Center 'Users' page. On the left is a navigation sidebar with options like Dashboard, Users, Groups, Settings, Multi-account permissions, and Application assignments. The main content area shows a table of users. At the top of the table, there are buttons for 'Delete users' and 'Manage sync', and a search bar labeled 'Find users'. The table has four columns: Username, Display name, Status, and MFA device. There are four rows of users, each with a checkbox in the first column. The users are: user1@corp.pcluster.com, admin1@corp.pcluster.com, user2@corp.pcluster.com, and admin2@corp.pcluster.com. All users have a status of 'Enabled' and an MFA device of 'None'.

| <input type="checkbox"/> | Username                 | Display name             | Status  | MFA device |
|--------------------------|--------------------------|--------------------------|---------|------------|
| <input type="checkbox"/> | user1@corp.pcluster.com  | user1@corp.pcluster.com  | Enabled | None       |
| <input type="checkbox"/> | admin1@corp.pcluster.com | admin1@corp.pcluster.com | Enabled | None       |
| <input type="checkbox"/> | user2@corp.pcluster.com  | user2@corp.pcluster.com  | Enabled | None       |
| <input type="checkbox"/> | admin2@corp.pcluster.com | admin2@corp.pcluster.com | Enabled | None       |

## 將您的應用程式新增至 IAM 身分中心

將使用者與 IAM 身分中心同步後，您將需要新增應用程式。這樣可以從 IAM 身分中心入口網站設定哪些啟用 SSO 的應用程式。在這種情況下，我們將添加 AWS ParallelCluster UI 作為應用程序，而 IAM 身分中心將成為身份提供者。

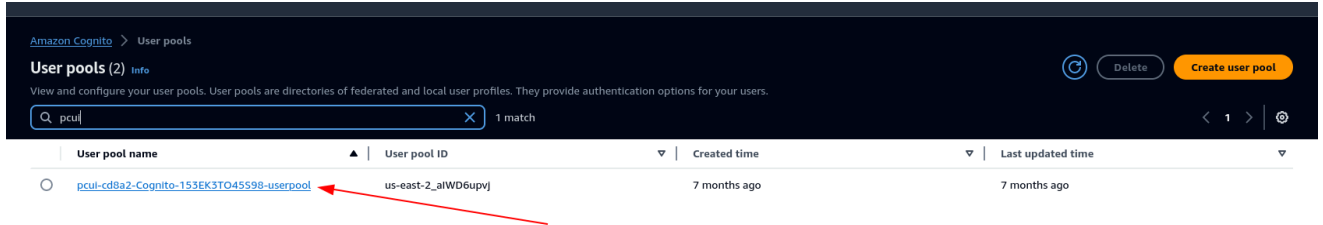
下一步會將 AWS ParallelCluster UI 新增為 IAM 身分中心中的應用程式。AWS ParallelClusterUI 是一個 Web 門戶，可幫助用戶管理其集群。如需詳細資訊，請參閱 [AWS ParallelClusterUI](#)。

在身分識別中心中設定應用程式

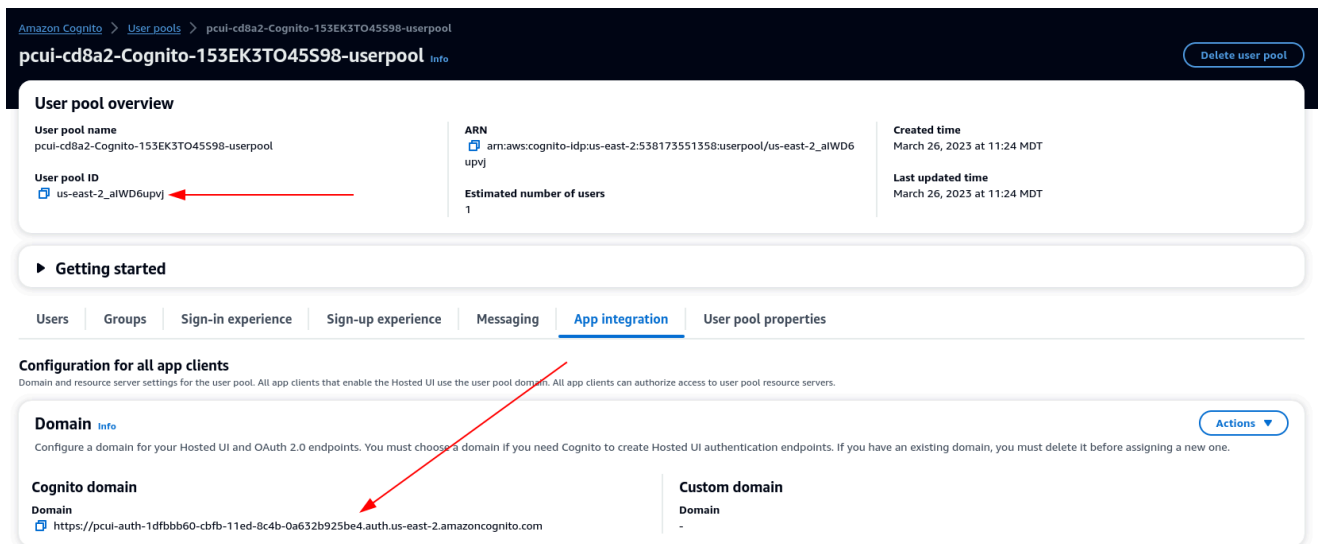
1. 在 IAM 身分中心 > 應用程式下 (位於左側功能表列中，按一下 [應用程式])
2. 按一下新增應用
3. 選擇添加自定義 SAML 2.0 應用程序
4. 點擊下一步
5. 選取您要使用的顯示名稱和說明 (例如 PCUI 和 AWS ParallelCluster UI)
6. 在 IAM 身分中心中繼資料下，複製 IAM 身分中心 SAML 中繼資料檔案的連結，然後儲存以供稍後使用，這將在 Web 應用程式上設定 SSO 時使用
7. 在「應用程式內容」下的「應用程式啟動 URL」中，輸入您的 PCUI 位址。這可以通過轉到 CloudFormation 控制台，選擇與 PCUI 對應的堆棧 (例如，並行 Icluster-ui)，然後轉到「輸出」選項卡以查找 UIURL ParallelCluster

例如：<https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com>

8. 在 [應用程式中繼資料] 下方，選擇 [手動輸入您的] 然後提供下列值。
  - a. 重要事項：請務必將網域前置詞、區域和使用者集區識別碼值取代為您環境的特定資訊。
  - b. 透過開啟 Amazon Cognito > 使用者集區主控台，即可取得網域前置詞、區域和使用者集區識別碼



- c. 選取對應至 PCUI 的使用者集區 (其中會有一個使用者集區名稱，如 PCUI CD8A2-認知系統-153EK3TO45S98-使用者集區)
  - d. 導覽至應用程式整合



9. <domain-prefix>應用程式宣告取用者服務 (ACS) 網址：  
 <region>. 亞馬遜認識.com /三個/識別  
 應用 SAML 受眾:龔:亞馬遜:齒:SP: <userpool-id>
10. 選擇提交。然後，前往您新增之應用程式的「詳細資料」頁面。
11. 選取動作下拉式清單，然後選擇編輯屬性對應。然後，提供下列屬性。
  - a. 應用程式中的用戶屬性：主題 (注意：主題已預填充。) → 映射到此字符串值或 IAM 身份中心的用戶屬性：\${用戶：電子郵件}，格式：電子郵件地址

- b. 應用程式中的用戶屬性：電子郵件 → 映射到此字符串值或 IAM 身份中心的用戶屬性：\${用戶：電子郵件}，格式：未指定

Attribute mappings for PCUI

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. [Learn more](#)

| User attribute in the application | Maps to this string value or user attribute in IAM Identity Center | Format       |        |
|-----------------------------------|--------------------------------------------------------------------|--------------|--------|
| Subject                           | \$(user:email)                                                     | emailAddress |        |
| email                             | \$(user:email)                                                     | unspecified  | Remove |

[Add new attribute mapping](#)

[Cancel](#) [Save changes](#)

12. 儲存您的變更。

13. 選擇「指派使用者」按鈕，然後將您的使用者指派給應用程式。這些是作用中目錄中將有權存取 PCUI 介面的使用者。

IAM Identity Center > Applications > PCUI

### PCUI

Details Actions

| Icon | Display name | Description            |
|------|--------------|------------------------|
|      | PCUI         | AWS ParallelCluster UI |

Assigned users (1) Remove access Assign Users

The following users and groups from your connected directory can access this application. [Learn more](#)

Search for an assigned user or group

| User/Group name                                                   | Type |
|-------------------------------------------------------------------|------|
| <input type="checkbox"/> <a href="#">admin1@corp.pcluster.com</a> | User |

在您的使用者集區中將 IAM 身分中心設定為 SAML IdP

1. 在使用者集區設定中，選取 [登入體驗] > [新增身分提供者]

The screenshot shows the AWS IAM console interface for a Cognito user pool. The breadcrumb navigation is: Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool. The user pool name is 'parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool' and the ID is 'us-east-1\_Bgyu7LLz6'. The ARN is 'arn:aws:cognito-idp:us-east-1:538173551358:userpool/us-east-1\_Bgyu7LLz6'. The estimated number of users is 1. The user pool was created on November 5, 2023 at 12:32 MST and last updated on the same date and time. The 'Getting started' section has tabs for Users, Groups, Sign-in experience (selected), Sign-up experience, Messaging, App integration, and User pool properties. The 'Cognito user pool sign-in' section explains that users can sign in using email, phone number, or user name. The 'Federated identity provider sign-in (0)' section shows a table with columns for Identity provider, Identity provider type, Created time, and Last updated time. A red arrow points to the 'Delete' button for an identity provider.

2. 選擇一個 SAML IdP
3. 對於提供者名稱提供 IdentityCenter
4. 在「中繼資料文件來源」下選擇「輸入中繼資料文件端點 URL」，並提供 Identity Center 應用程式設定期間複製
5. 在屬性下，對於電子郵件選擇電子郵件



**SAML**  
Configure a SAML 2.0 identity provider for your user pool.

**Register your app with your SAML provider**  
To connect a SAML provider to Cognito, add your user pool as a relying party or application with your SAML 2.0 identity provider, and upload a metadata document to Cognito.

**Set up SAML federation with this user pool**

**Provider name** [Info](#)  
Enter a friendly name for your SAML 2.0 identity provider.

**Identifiers - optional** [Info](#)  
Enter identifiers for this provider. Identifiers can be used to redirect users to the correct IdP in multitenant apps.

Separate each identifier by a comma

**Sign-out flow** [Info](#)  
 Add sign-out flow  
Enable simultaneous sign-out from the SAML provider and Cognito.

**Metadata document source** [Info](#)  
Provide a SAML metadata document. This document is issued by your SAML provider. It includes the issuer's name, expiration information, and keys that can be used to validate the response from the identity provider.  
 Upload metadata document  
 Enter metadata document endpoint URL

**Enter metadata document endpoint URL** [Info](#)

**Map attributes between your SAML provider and your user pool** [Info](#)  
Your required attributes are mapped to the equivalent SAML attributes. Each attribute you add must be mapped to a SAML attribute.

| User pool attribute | SAML attribute |
|---------------------|----------------|
| email               | email          |

[Add another attribute](#)

[Cancel](#) [Add identity provider](#)

## 6. 選取 新增身分供應商。

### 將 IdP 與使用者集區應用程式用戶端整合

1. 接下來，在用戶池的「應用程式集成」部分下，選擇「應用程式客戶端列表」下列出的客戶端

Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool

## parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool Info

[Delete user pool](#)

### User pool overview

|                                                                           |                                                                                      |                                                           |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>User pool name</b><br>parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool | <b>ARN</b><br>arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_Bgyu7Lz6 | <b>Created time</b><br>November 5, 2023 at 12:32 MST      |
| <b>User pool ID</b><br>us-east-1_Bgyu7Lz6                                 | <b>Estimated number of users</b><br>1                                                | <b>Last updated time</b><br>November 5, 2023 at 12:32 MST |

### Getting started

Users | Groups | Sign-in experience | Sign-up experience | Messaging | **App integration** | User pool properties

### Configuration for all app clients

Domain and resource server settings for the user pool. All app clients that enable the Hosted UI use the user pool domain. All app clients can authorize access to user pool resource servers.

#### Domain Info

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. You must choose a domain if you need Cognito to create Hosted UI authentication endpoints. If you have an existing domain, you must delete it before assigning a new one.

[Actions](#)

|                                                                                                          |                      |
|----------------------------------------------------------------------------------------------------------|----------------------|
| <b>Cognito domain</b>                                                                                    | <b>Custom domain</b> |
| <b>Domain</b><br>https://pcul-auth-06f29200-7c12-11ee-b755-0e11297ecb0d.auth.us-east-1.amazoncognito.com | <b>Domain</b><br>-   |

#### Resource servers (0) Info

Configure resource servers. A resource server is a remote server that authorizes access based on OAuth 2.0 scopes in an access token.

[Edit](#) [Delete](#) [Create resource server](#)

| Resource server name | Resource server identifier | Custom scopes |
|----------------------|----------------------------|---------------|
| No resource servers  |                            |               |

[Create resource server](#)

#### App client defaults

Hosted UI customization and advanced security settings for the user pool. You can customize the Hosted UI and advanced security in app clients to override the defaults.

#### Hosted UI customization Info

Customize the hosted sign-up and sign-in pages to match your app's style and branding by uploading your own logo and customized CSS.

[Edit](#)

|                  |                        |
|------------------|------------------------|
| <b>Logo</b><br>- | <b>Custom CSS</b><br>- |
|------------------|------------------------|

#### Advanced security Info

Configure advanced security features, including Cognito's automatic responses to suspicious user activity. Advanced security adds cost to your bill. [See pricing](#)

[Enable](#)

**Status**  
 Disabled

#### App client list

The app clients that integrate your apps with your user pool. Configure client overrides to user pool default configurations, and configure Amazon Pinpoint analytics.

#### App clients and analytics (1) Info

Configure an app client. App clients are the user pool authentication resources attached to your app. Select an app client to configure the permitted authentication actions for an app.

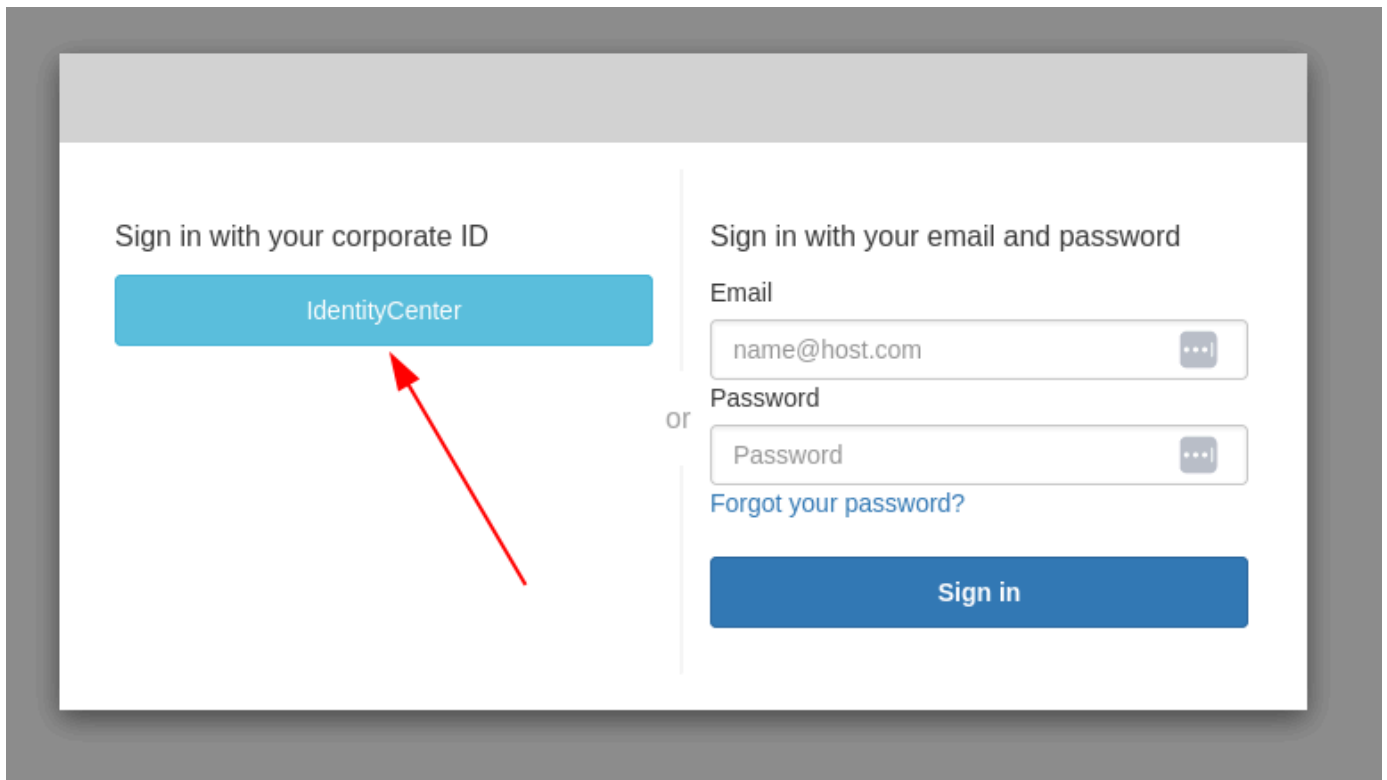
[Refresh](#) [Delete](#) [Create app client](#)

| App client name                                                     | Client ID |
|---------------------------------------------------------------------|-----------|
| <input type="radio"/> <a href="#">CognitoAppClient-ETqXbqI5wRVs</a> | ...       |

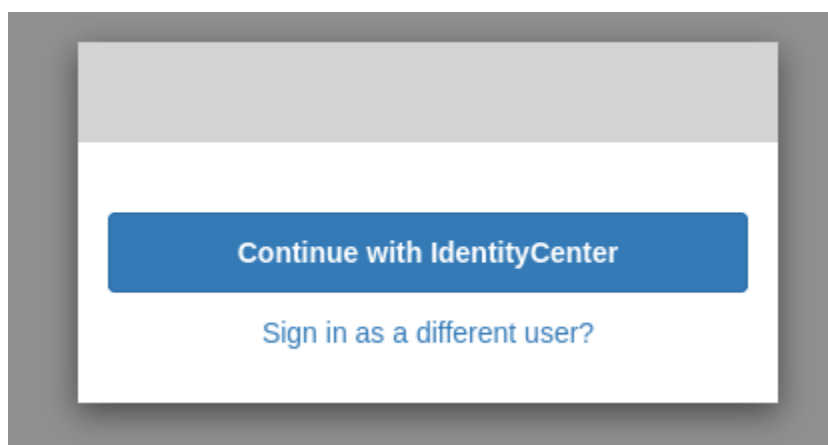
2. 在託管 UI 下，選擇編輯
3. 在「身份提供者」下也IdentityCenter可以選擇。
4. 選擇 Save changes (儲存變更)

## 驗證您的設定

1. 接下來，我們將透過登入 PCUI 來驗證剛剛建立的設定。登入您的 PCUI 入口網站，您現在應該會看到使用您的公司 ID 登入的選項：



2. 按一下 IdentityCenter 按鈕應會將您導向 IAM 身分中心 IdP 登入，然後接著一個包含您應用程式 (包括 PCUI) 的頁面，開啟該應用程式。
3. 進入下列畫面後，您的使用者就會新增至 Cognito 使用者集區。



### 讓您的使用者成為管理員

1. 現在導航到 Amazon Cognito > 用戶池控制台，然後選擇新創建的用戶，該用戶應該具有前綴身份中心

The screenshot shows the AWS IAM console interface for a user pool. The top navigation bar includes 'Amazon Cognito' > 'User pools' > 'parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool'. The main content area is titled 'parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool' and includes a 'Delete user pool' button.

**User pool overview**

|                                                                           |                                                                                      |                                                           |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>User pool name</b><br>parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool | <b>ARN</b><br>arn:aws:cognito-idp:us-east-1:000000000000:userpool/us-east-1_Bgyu7Lz6 | <b>Created time</b><br>November 5, 2023 at 12:32 MST      |
| <b>User pool ID</b><br>us-east-1_Bgyu7LLz6                                | <b>Estimated number of users</b><br>2                                                | <b>Last updated time</b><br>November 5, 2023 at 12:32 MST |

**Getting started**

Users | Groups | Sign-in experience | Sign-up experience | Messaging | App integration | User pool properties

**Users (2)** Info

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

Property:

| User name                                                      | Email address            | Email verified | Confirmation status | Status  |
|----------------------------------------------------------------|--------------------------|----------------|---------------------|---------|
| <input type="radio"/> c8f7eccc-e647-4227-afa2-080274ebfefe     | user@amazon.com          | Yes            | Confirmed           | Enabled |
| <input type="radio"/> identitycenter_admin1@corp.pcluster.c... | admin1@corp.pcluster.com | No             | External provider   | Enabled |

**Import users (0)** Info

View and create user CSV Import jobs. Amazon Cognito can import users into this user pool from a specially-formatted CSV file. You can't import user passwords.

| Job name                  | Status | Imported users | Skipped users | Failed users | CloudWatch logs | Created time |
|---------------------------|--------|----------------|---------------|--------------|-----------------|--------------|
| No user import jobs found |        |                |               |              |                 |              |

2. 在 [群組成員資格] 下方，選取 [新增使用者至群組]，選擇 [管理員]，
3. 現在，當您單擊繼續時，IdentityCenter 您將被導航到 AWS ParallelCluster UI 頁面。

# AWS ParallelCluster 疑難排

AWS ParallelCluster 社群會維護一個 Wiki 頁面，提供許多 [Wi AWS ParallelCluster GitHub ki](#) 上的疑難排解提示。如需已知問題的清單，請參閱[已知問題](#)。

## 主題

- [嘗試建立叢集](#)
- [嘗試執行工作](#)
- [嘗試更新叢集](#)
- [嘗試訪問存儲](#)
- [嘗試刪除叢集](#)
- [嘗試升級 AWS ParallelCluster API 堆棧](#)
- [在計算節點初始化中看到錯誤](#)
- [疑難排解叢集健康度](#)
- [疑難排解叢集部署問](#)
- [使用地形進行叢集部署疑難排解](#)
- [排解擴展問題](#)
- [放置群組和執行個體啟動問題](#)
- [無法取代的目錄](#)
- [疑難排解 NICE DCV 中的問題](#)
- [使用整合進行叢 AWS Batch 集的疑難排解](#)
- [疑難排解多使用者與活動目錄整合](#)
- [排解自訂 AMI 問題](#)
- [疑難排解cfn-hup未執行時的叢集更新逾時](#)
- [網路疑難排](#)
- [onNodeUpdated自訂動作的叢集更新失敗](#)
- [查看自定義配Slurm置的錯誤](#)
- [叢集警示](#)
- [其他支援](#)

## 嘗試建立叢集

使用 AWS ParallelCluster 版本 3.5.0 及更新版本建立叢集，且叢集建立失敗且 `--rollback-on-failure` 設為 `false`，請使用 `pcluster describe-cluster` CLI 命令取得狀態和失敗資訊。在這種情況下，預期 `clusterStatus` 的 `pcluster describe-cluster` 輸出是 `CREATE_FAILED`。檢查輸出中的 `failures` 部分以找到 `failureCode` 和 `failureReason`。然後，在以下部分中，找到相符 `failureCode` 的其他疑難排解說明。如需詳細資訊，請參閱 [pcluster describe-cluster](#)。

在下列各節中，我們建議您檢查 `head` 節點上的記錄檔，例如 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案。如需有關 AWS ParallelCluster 記錄檔及其檢視方式的詳細資訊，請參閱 [調試的關鍵日誌](#) 和 [擷取和保留記錄](#)。

如果您沒有 `failureCode`，請瀏覽至 AWS CloudFormation 主控台以檢視叢集堆疊。檢查其他資源的 `HeadNodeWaitCondition` 或失敗，以尋找其他失敗詳細資訊。Status Reason 如需詳細資訊，請參閱 [查看 AWS CloudFormation 事件 CREATE\\_FAILED](#)。檢查頭節點上的 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 文件。

### failureCode 是 OnNodeConfiguredExecutionFailure

- 為什麼會失敗？

您在組態中 `OnNodeConfigured` 的 `head` 節點區段中提供了自訂指令碼，以建立叢集。但是，自訂指令碼無法執行。

- 如何解決？

檢查 `/var/log/cfn-init.log` 檔案以進一步瞭解失敗，以及如何修正自訂指令碼中的問題。接近此記錄檔結尾時，您可能會在訊息後看到與 `OnNodeConfigured` 指令碼相關的執行資訊 `Running command runpostinstall` 訊。

### failureCode 是 OnNodeConfiguredDownloadFailure

- 為什麼會失敗？

您在組態中 `OnNodeConfigured` 的 `head` 節點區段中提供了自訂指令碼，以建立叢集。不過，自訂指令碼無法下載。

- 如何解決？

請確定 URL 有效，而且存取權設定正確。如需有關自訂啟動程序指令碼組態的詳細資訊，請參閱[自訂引導動作](#)。

檢查/var/log/cfn-init.log檔案。接近此記錄檔結尾時，您可能會在訊Running command runpostinstall息後看到與OnNodeConfigured指令碼處理相關的執行資訊，包括下載。

## failureCode 是 OnNodeConfiguredFailure

- 為什麼會失敗？

您在組態中OnNodeConfigured的 head 節點區段中提供了自訂指令碼，以建立叢集。不過，叢集部署中使用自訂指令碼失敗。無法確定立即原因，需要進行額外的調查。

- 如何解決？

檢查/var/log/cfn-init.log檔案。接近此記錄檔結尾時，您可能會在訊息後看到與OnNodeConfigured指令碼處理相關的執行資Running command runpostinstall訊。

## failureCode 是 OnNodeStartExecutionFailure

- 為什麼會失敗？

您在組態中OnNodeStart的 head 節點區段中提供了自訂指令碼，以建立叢集。但是，自訂指令碼無法執行。

- 如何解決？

檢查/var/log/cfn-init.log檔案以進一步瞭解失敗，以及如何修正自訂指令碼中的問題。接近此記錄檔結尾時，您可能會在訊息後看到與OnNodeStart指令碼相關的執行資Running command runpreinstall訊。

## failureCode 是 OnNodeStartDownloadFailure

- 為什麼會失敗？

您在組態中OnNodeStart的 head 節點區段中提供了自訂指令碼，以建立叢集。不過，自訂指令碼無法下載。

- 如何解決？

請確定 URL 有效，而且存取權設定正確。如需有關自訂啟動程序指令碼組態的詳細資訊，請參閱[自訂引導動作](#)。

檢查/var/log/cfn-init.log檔案。接近此記錄檔結尾時，您可能會在訊Running command runpreinstall息後看到與OnNodeStart指令碼處理相關的執行資訊，包括下載。

## failureCode 是 OnNodeStartFailure

- 為什麼會失敗？

您在組態中OnNodeStart的 head 節點區段中提供了自訂指令碼，以建立叢集。不過，叢集部署中使用自訂指令碼失敗。無法確定立即原因，需要進行額外的調查。

- 如何解決？

檢查/var/log/cfn-init.log檔案。接近此記錄檔結尾時，您可能會在訊息後看到與OnNodeStart指令碼處理相關的執行資Running command runpreinstall訊。

## failureCode 是 EbsMountFailure

- 為什麼會失敗？

叢集配置中定義的 EBS 磁碟區無法掛載。

- 如何解決？

檢查/var/log/chef-client.log檔案中的失敗詳細資訊。

## failureCode 是 EfsMountFailure

- 為什麼會失敗？

叢集組態中定義的 Amazon EFS 磁碟區無法掛接。

- 如何解決？

如果您定義了現有的 Amazon EFS 檔案系統，請確定叢集和檔案系統之間允許流量。如需詳細資訊，請參閱 [SharedStorageEfsSettings//FileSystemId](#)。

檢查/var/log/chef-client.log檔案中的失敗詳細資訊。



## failureCode 是 FsxMountFailure

- 為什麼會失敗？

叢集組態中定義的 Amazon FSx 檔案系統無法掛載。

- 如何解決？

如果您定義了現有的 Amazon FSx 檔案系統，請確定叢集和檔案系統之間允許流量。如需詳細資訊，請參閱 [SharedStorageFsxLustreSettings//FileSystemId](#)。

檢查 `/var/log/chef-client.log` 檔案中的失敗詳細資訊。

## failureCode 是 RaidMountFailure

- 為什麼會失敗？

叢集配置中定義的 RAID 卷冊無法掛載。

- 如何解決？

檢查 `/var/log/chef-client.log` 檔案中的失敗詳細資訊。

## failureCode 是 AmiVersionMismatch

- 為什麼會失敗？

用於建立自訂 AMI 的 AWS ParallelCluster 版本與用於設定叢集的 AWS ParallelCluster 版本不同。在 CloudFormation 主控台中，檢視叢集 CloudFormation 堆疊詳細資料，並檢查 `HeadNodeWaitCondition` 以取得有關 AWS ParallelCluster 版本和 AMI 的其他詳細資料。Status Reason 如需詳細資訊，請參閱 [查看 AWS CloudFormation 事件 CREATE\\_FAILED](#)。

- 如何解決？

確保用於創建自定義 AMI 的 AWS ParallelCluster 版本與用於配置集群的 AWS ParallelCluster 版本相同。您可以變更自訂 AMI 版本或 `pcluster` CLI 版本以使其相同。

## failureCode 是 InvalidAmi

- 為什麼會失敗？

自定義 AMI 無效，因為它不是使用 AWS ParallelCluster。

- 如何解決？

使用 `pcluster build-image` 指令將 AMI 設為父映像來建立 AMI。如需詳細資訊，請參閱 [pcluster build-image](#)。

## **failureCodefailureReason**無法設置頭節點。**HeadNodeBootstrapFailure**

- 為什麼會失敗？

無法確定立即原因，需要進行額外的調查。例如，可能是叢集處於受保護狀態，這可能是因為無法佈建靜態運算叢集所造成。

- 如何解決？

檢查 `/var/log/chef-client.log` 檔案中的失敗詳細資訊。

### Note

如果您看到 `RuntimeError` 例外狀況 `Cluster state has been set to PROTECTED mode due to failures detected in static node provisioning`，表示叢集處於受保護狀態。如需詳細資訊，請參閱 [如何調試受保護模式](#)。

## **failureCodefailureReason**叢集建立逾時。**HeadNodeBootstrapFailure**

- 為什麼會失敗？

根據預設，叢集建立完成的時間限制為 30 分鐘。如果叢集建立尚未在此時間範圍內完成，叢集建立會失敗，並顯示逾時錯誤。叢集建立可能會因為不同的原因而逾時。例如，逾時失敗可能是由於頭節點建立失敗、網路問題、在頭節點執行時間過長的自訂指令碼、在計算節點中執行的自訂指令碼中產生錯誤，或是計算節點佈建的等待時間過長。無法確定立即原因，需要進行額外的調查。

- 如何解決？

檢查 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案以取得失敗詳細資訊。如需有關 AWS ParallelCluster 記錄檔及如何取得記錄檔的詳細資訊，請參閱 [調試的關鍵日誌](#) 和 [擷取和保留記錄](#)。

您可能會在這些日誌中發現以下內容。

- 看到 **Waiting for static fleet capacity provisioning** 附近的盡頭 **chef-client.log**

這表示叢集建立在等待靜態節點開機時逾時。如需詳細資訊，請參閱 [在計算節點初始化中看到錯誤](#)。

- 查看 **OnNodeConfigured** 或 **OnNodeStart** 頭節點腳本未在結束時完成 **cfn-init.log**

這表示 **OnNodeConfigured** 或 **OnNodeStart** 自訂指令碼需要很長時間才能執行，並造成逾時錯誤。檢查您的自定義腳本是否存在可能導致其長時間運行的問題。如果您的自訂指令碼需要很長時間才能執行，請考慮將 `DevSettings` 區段新增至叢集配置檔，以變更逾時限制，如下列範例所示：

```
DevSettings:
  Timeouts:
    HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- 找不到日誌，或者頭節點未成功創建

頭節點可能未成功創建，並且無法找到日誌。在 CloudFormation 主控台中，檢視叢集堆疊詳細資料，以檢查其他失敗詳細資料。

## **failureCodefailureReason**無法啟動頭節點。**HeadNodeBootstrapFailure**

- 為什麼會失敗？

無法確定立即原因，需要進行額外的調查。

- 如何解決？

檢查 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案。

## failureCode 是 ResourceCreationFailure

- 為什麼會失敗？

在叢集建立程序期間，建立部分資源失敗。失敗可能由於各種原因而發生。例如，資源建立失敗可能是容量問題或 IAM 政策設定錯誤所造成。

- 如何解決？

在 CloudFormation 主控台中，檢視叢集堆疊，以檢查其他資源建立失敗詳細資訊。

## failureCode 是 ClusterCreationFailure

- 為什麼會失敗？

無法確定立即原因，需要進行額外的調查。

- 如何解決？

在 CloudFormation 主控台中，檢視叢集堆疊並檢查以 Status Reason 尋找其他失敗詳細資訊。HeadNodeWaitCondition

檢查 /var/log/cfn-init.log 和 /var/log/chef-client.log 檔案。

看到 CloudFormation , **WaitCondition timed out...** 在, 堆

如需詳細資訊，請參閱 [failureCode failureReason 叢集建立逾時。HeadNodeBootstrapFailure](#)。

看到 CloudFormation , **Resource creation cancelled** 在, 堆

如需詳細資訊，請參閱 [failureCode 是 ResourceCreationFailure](#)。

查看 **Failed to run cfn-init...** 或 AWS CloudFormation 堆棧中的其他錯誤

檢查 /var/log/cfn-init.log 和以取 /var/log/chef-client.log 得其他失敗詳細資訊。

## 看到chef-client.log結束 INFO: Waiting for static fleet capacity provisioning

這與等待靜態節點開機時的叢集建立逾時有關。如需詳細資訊，請參閱 [在計算節點初始化中看到錯誤](#)。

## 看得見 Failed to run preinstall or postinstall in cfn-init.log

您在叢集配置HeadNode區段中有OnNodeConfigured或OnNodeStart指令碼。指令碼無法正常運作。檢查/var/log/cfn-init.log檔案中的自訂指令碼錯誤詳細資訊。

## 看到 CloudFormation , This AMI was created with xxx, but is trying to be used with xxx... 在, 堆

如需詳細資訊，請參閱 [failureCode 是 AmiVersionMismatch](#)。

## 看到 CloudFormation , This AMI was not baked by AWS ParallelCluster... 在, 堆

如需詳細資訊，請參閱 [failureCode 是 InvalidAmi](#)。

## 看到pcluster create-cluster命令無法在本地運行

請~/parallelcluster/pcluster-cli.log在本機檔案系統中檢查失敗詳細資訊。

## 其他支援

請遵循中的疑難排解指引[疑難排解叢集部署問](#)。

請檢查您的案例是否涵蓋在的「[GitHub 已知問題](#)」AWS ParallelCluster 中 GitHub。

如需其他支援，請參閱[其他支援](#)。

## 嘗試執行工作

### **srun** 互動式工作因錯誤而失敗 **srun: error: fwd\_tree\_thread: can't find address for <host>, check slurm.conf**

- 為什麼會失敗？

您執行命 `srun` 來提交工作，然後使用命 `pcluster update-cluster` 來增加佇列的大小，而不必在更新完成後重新啟動 Slurm 精靈。

Slurm 在樹狀結構中組織 Slurm 守護進程以優化通信。只有在守護進程啟動時才會更新此層次結構。

假設您使 `srun` 用啟動工作，然後執行 `pcluster update-cluster` 命令來增加佇列的大小。新的運算節點會在更新過程中啟動。然後，將您的工作 Slurm 排入其中一個新的計算節點。在這種情況下，Slurm 守護進程和 `srun` 不檢測新的計算節點。`srun` 返回一個錯誤，因為它沒有檢測到新的節點。

- 如何解決？

重新啟動所有計算節點上的 Slurm 精靈，然後使用 `srun` 來提交工作。您可以執行重新啟動計算節點的 `scontrol reboot` 命令來排程 Slurm 精靈重新啟動。如需詳細資訊，請參閱 Slurm 文件中的 [scontrol 重新開機](#)。您也可以要求重新啟動對應的 `systemd` 服務，以手動重新啟動計算節點上的 Slurm 精靈。

## 使用 `squeue` 命令將 Job 停留在 CF 狀態

這可能是動態節點啟動的問題。如需詳細資訊，請參閱 [在計算節點初始化中看到錯誤](#)。

### **執行大規模工作並查看 `nfsd: too many open connections, consider increasing the number of threads in /var/log/messages`**

使用網路檔案系統時，當達到網路限制時，I/O 等待時間也會增加。這可能會導致軟鎖定，因為網路是用來寫入網路和 I/O 指標的資料。

在第 5 代執行個體中，我們使用 ENA 驅動程式公開封包計數器。這些計數器會計算網路達到執行個體頻寬限制 AWS 時所形成的封包。您可以檢查這些計數器，看看它們是否大於 0。如果是這樣，那麼您已經超過了帶寬限制。您可以透過執行來檢視這些計數器 `ethtool -S eth0 | grep exceeded`。

超過網路限制通常是因為支援過多 NFS 連線所致。當您達到或超過網路限制時，這是首先要檢查的事情之一。

例如，下列輸出會顯示捨棄的套件：

```
$ ethtool -S eth0 | grep exceeded
bw_in_allowance_exceeded: 38750610
bw_out_allowance_exceeded: 1165693
pps_allowance_exceeded: 103
conntrack_allowance_exceeded: 0
linklocal_allowance_exceeded: 0
```

若要避免收到此訊息，請考慮將 head 節點執行個體類型變更為效能更高的執行個體類型。請考慮將資料儲存移至不是以 NFS 共用形式匯出的共用儲存檔案系統，例如 Amazon EFS 或 Amazon FSx。如需詳細資訊[共用儲存](#)，請參閱 AWS ParallelCluster Wiki 上的和[最佳實務](#) GitHub。

## 執行 MPI 工作

### 啟用除錯模式

若要啟用 OpenMPI 除錯模式，請參閱[開啟 MPI 有哪些控制項可協助除錯](#)。

若要啟用 IntelMPI 除錯模式，請參閱[其他環境變數](#)。

### 在工作輸出 **OPAL ERROR** 中查看 **MPI\_ERRORS\_ARE\_FATAL** 和

這些錯誤代碼來自應用程式中的 MPI 層。若要瞭解如何從應用程式取得 MPI 偵錯記錄檔，請參閱[啟用除錯模式](#)。

此錯誤的可能原因是您的應用程式已針對特定 MPI 實作而編譯，例如 OpenMPI，而您嘗試使用不同的 MPI 實作來執行它，例如 IntelMPI。確保您使用相同的 MPI 實現編譯和運行應用程式。

### 禁用託管 DNS **mpirun** 的情況下使用

對於使用 [SlurmSettings/Ds/ DisableManagedDns](#) 和 [UseEc2 主機名稱](#) 設定為建立的叢集 true，DNS 無法解析 Slurm 節點名稱。Slurm 可以在 nodenames 未啟用時啟動 MPI 處理程序，以及 MPI 工作是否在內容中執行。Slurm 我們建議按照 [SlurmMPI 用戶指南中的指引](#) 來執行 MPI 任務。Slurm

## 嘗試更新叢集

### `pcluster update-cluster`命令無法在本地運行

請在 `~/.parallelcluster/pcluster-cli.log` 在本機檔案系統中檢查失敗詳細資訊。

### 看到 `clusterStatus` 是 `UPDATE_FAILED` 用 `pcluster describe-cluster` 命令

如果叢集堆疊更新還原，請檢查 `/var/log/chef-client.logs` 檔案以取得錯誤詳細資訊。

請查看的「已知問題」中是否 [GitHub 已提及您 AWS ParallelCluster 的問題](#) GitHub。

## 叢集更新逾時

這可能是與 `cfn-hup` 未執行有關的問題。如果 `cfn-hup` 惡魔因外部原因終止，則不會自動重新啟動。如果 `cfn-hup` 未執行，則在叢集更新期間，CloudFormation 堆疊會如預期般啟動更新程序，但不會在頭節點上啟動更新程序，且堆疊部署最終會逾時。如需詳細資訊，請參閱 [疑難排解 `cfn-hup` 未執行時的叢集更新逾時](#) 疑難排解並從問題中復原。

## 嘗試訪問存儲

### 使用外部 Amazon FSx for Lustre 文件系統

請確定叢集與檔案系統之間允許流量。檔案系統必須與允許透過連接埠 988、1021、1022 和 1023 之輸入和輸出 TCP 流量的安全性群組相關聯。如需如何設定安全性群組的詳細資訊，請參閱 [FileSystemId](#)。

### 使用外部 Amazon Elastic File System 檔案系統

請確定叢集與檔案系統之間允許流量。檔案系統必須與允許透過連接埠 988、1021、1022 和 1023 之輸入和輸出 TCP 流量的安全性群組相關聯。如需如何設定安全性群組的詳細資訊，請參閱 [FileSystemId](#)。

## 嘗試刪除叢集

### 該 `pcluster delete-cluster` 命令無法在本地運行

檢查本機 `~/.parallelcluster/pcluster-cli.log` 檔案系統中的檔案。



## 叢集堆疊無法刪除

如果叢集堆疊無法刪除，請檢查 CloudFormation 堆疊事件訊息。

請在的「已知問題」中檢查您的[問題是否GitHub 已提及](#) GitHub。AWS ParallelCluster

## 嘗試升級 AWS ParallelCluster API 堆棧

請在的「已知問題」中檢查您的[問題是否GitHub 已提及](#) GitHub。AWS ParallelCluster

## 在計算節點初始化中看到錯誤

### 看到Node bootstrap error在 clustermgtd.log

這個問題與計算節點無法引導有關。如需有關如何偵錯受叢集保護模式問題的資訊，請參閱[如何調試受保護模式](#)。

### 我設定了隨需容量保留 (ODC) 或區域預留執行個體

ODC，其中包含具有多個網路介面的執行個體，例如 P4d、P4de 和特雷尼姆 (Trn) AWS

在叢集配置檔中，檢查HeadNode是否位於公用子網路中，以及計算節點是否位於私有子網路中。

ODC 是目標性的 OCRS

**Unable to read file '/opt/slurm/etc/pcluster/run\_instances\_overrides.json'.**即使我已經按照/opt/slurm/etc/pcluster/run\_instances\_overrides.json中給出的說明進行操作，也可以看到 [使用隨需容量保留 \(ODCR\) 啟動執行個體](#)

如果您將 3.1.1 到 3.2.1 AWS ParallelCluster 版與目標 OCRs 搭配使用，而且您也使用[執行個體覆寫 JSON 檔案](#)，則可能是您沒有正確格式化 JSON 檔案。您可能會在中看到錯誤clustermgtd.log，如下所示：

```
Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'.
Using default: {} in /var/log/parallelcluster/clustermgtd.
```

執行下列命令來驗證 JSON 檔案格式是否正確：

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

查看**Found RunInstances parameters override**.叢集**clustermgtd.log**建立失敗或執行作業失敗**slurm\_resume.log**時

如果您使用[執行個體覆寫 JSON 檔案](#)，請檢查檔案中是否正確設定佇列名稱和計算資源名稱。`/opt/slurm/etc/pcluster/run_instances_overrides.json`

看到**An error occurred (InsufficientInstanceCapacity)**我何**slurm\_resume.log**時無法運行作業，或者**clustermgtd.log**當我無法創建集群時

使用 PG-ODCR (放置組)

使用關聯的放置群組建立 ODCR 時，組態檔案中必須使用相同的放置群組名稱。在叢集配置中設定對應的放置群組名稱。

使用區域預留執行個體

如果您在叢集配置**true**中將區域預留執行個體與**PlacementGroup/Enabledto** 搭配使用，則可能會看到錯誤訊息，如下所示：

```
We currently do not have sufficient trn1.32xlarge capacity in the Availability Zone you requested (us-east-1d). Our system will be working on provisioning additional capacity. You can currently get trn1.32xlarge capacity by not specifying an Availability Zone in your request or choosing us-east-1a, us-east-1b, us-east-1c, us-east-1e, us-east-1f.
```

您可能會看到這個問題，是因為區域預留執行個體並未放置在相同的 UC (或 Spine) 中，這可能會在使用置放群組時造成容量不足錯誤 (ICE)。您可以停用叢集配置中的 [PlacementGroup群組] 設定，以判斷叢集是否可以配置執行個體，以檢查此情況。

**An error occurred (VcpuLimitExceeded)**在我無法運行作業**slurm\_resume.log**時看到，或者在**clustermgtd.log**我無法創建集群時看到

針對您使用的特定 Amazon EC2 執行個體類型，檢查您帳戶的 vCPU 限制。如果您看到的 vCPUs 數量少於您要求的 vCPU 數量為零，請求提高限制。如需如何檢視目前限制和請求新限制的相關資訊，請參閱 [Amazon EC2 使用者指南中的 Amazon EC2 服務配額](#)。

**An error occurred (InsufficientInstanceCapacity)在我無法運行作業slurm\_resume.log時看到，或者在clustermgtd.log我無法創建集群時看到**

您遇到容量不足的問題。請按照 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>來解決問題。

## 查看節點DOWN處於狀態 Reason (Code:InsufficientInstanceCapacity)...

您遇到容量不足的問題。請按照 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>來解決問題。如需快速不足容量容錯移轉模式 AWS ParallelCluster的詳細資訊，請參閱。[Slurm叢集快速容量不足容錯移轉](#)

## 看到cannot change locale (en\_US.utf-8) because it has an invalid name在 slurm\_resume.log

如果您的yum安裝程序不成功，而且地區設定設定處於不一致的狀態，就可能會發生這種情況。例如，當使用者終止安裝程序時，可能會造成這個問題。

若要確認原因，請採取下列動作：

- 執行 `su - pcluster-admin`。

殼層會顯示錯誤，例如，`cannot change locale...no such file or directory`。

- 執行 `localedef --list`。

返回一個空列表或不包含默認語言環境。

- 使用 `yum history` 和檢查最後一個 `yum` 命令 `yum history info #ID`。最後一個 ID 是否有 `Return-Code: Success`？

如果最後一個 ID 沒有 `Return-Code: Success`，則安裝後指令碼可能無法成功執行。

若要修正此問題，請嘗試使用 `yum reinstall glibc-all-langpacks`。重建之後，如果問題已修正，則不 `su - pcluster-admin` 會顯示錯誤或警告。

## 以前的情況都不適用於我的情況

要疑難排解計算節點初始化問題，請參閱[解決節點初始化問題](#)。

請檢查您的案例是否涵蓋在的「[GitHub 已知問題](#)」AWS ParallelCluster 中 GitHub。

如需其他支援，請參閱[其他支援](#)。

## 疑難排解叢集健康度

從 3.6.0 AWS ParallelCluster 版開始，叢集運作狀態指標會新增至 AWS ParallelCluster Amazon CloudWatch 儀表板。在以下各節中，您可以瞭解儀表板健全狀況指標，以及疑難排解和解決問題時可採取的動作。

### 主題

- [查看執行處理啟動設定錯誤圖](#)
- [查看狀態不良的執行個體錯誤圖表](#)
- [查看運算叢集閒置時間圖表](#)

## 查看執行處理啟動設定錯誤圖

如果您在圖表中看到非零值，則Instance Provisioning Errors表示用於支援 slurm 節點的 Amazon EC2 執行個體無法在或 API 上啟動。CreateFleet RunInstance

## 看得見 IAMPolicyErrors

- 發生了什麼事？

許多執行個體無法啟動，這是因為權限不足且具有錯誤碼的原因UnauthorizedOperation。

- 如何解決？

如果您已設定自訂[InstanceRole](#)或 [InstanceProfile](#)，請檢查您的 IAM 政策，並確認您使用的是正確的登入資料。

檢查clustermgtd文件中的靜態節點錯誤詳細信息。檢查slurm\_resume.log文件中的動態節點錯誤詳細信息。使用詳細資料進一步瞭解必須新增的遺失權限。

## 看得見 VcpuLimitErrors

- 發生了什麼事？

AWS ParallelCluster 無法啟動執行個體，因為它達到您 AWS 帳戶 針對叢集運算節點設定的特定 Amazon EC2 執行個體類型的 vCPU 限制。

- 如何解決？

檢查 `clustermgtd` 檔案中是否有靜態節點的 `VcpuLimitExceeded` 錯誤，並簽入 `slurm_resume.log` 檔案中的動態節點以取得其他詳細資料。若要解決此問題，您可以要求提高 vCPU 限制。如需有關如何檢視目前限制和請求新限制的詳細資訊，請參閱 [Amazon 彈性運算雲端 Linux 執行個體使用者指南中的 Amazon 彈性運算雲端服務配額](#)。

## 看得見 VolumeLimitErrors

- 發生了什麼事？

您已達到您的 Amazon EBS 磁碟區限制 AWS 帳戶，且 AWS ParallelCluster 無法使用錯誤碼 `InsufficientVolumeCapacity` 或 `VolumeLimitExceeded` 啟動執行個體。

- 如何解決？

檢查 `clustermgtd` 檔案是否有靜態節點，並檢查 `slurm_resume.log` 檔案是否有動態節點，以取得額外的磁碟區限制詳細資訊。若要解決此問題，您可以使用不同的磁碟區 AWS 區域、清理現有磁碟區，或聯絡 Sup AWS port 中心提交要求以提高 Amazon EBS 磁碟區限制。

## 看得見 InsufficientCapacityErrors

- 發生了什麼事？

AWS ParallelCluster 沒有足夠的容量來啟動 Amazon EC2 執行個體來支援節點。

- 如何解決？

檢查 `clustermgtd` 文件中的靜態節點，並檢查 `slurm_resume.log` 文件中的動態節點以獲取容量不足的錯誤詳細信息。若要疑難排解此問題，請依照 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> 的指引進行。

## OtherInstanceLaunchFailures

- 發生了什麼事？

用於支援運算節點的 Amazon EC2 執行個體無法使用 `CreateFleet` 或 `RunInstance` API 啟動。

- 如何解決？

檢查 `clustermgtd` 文件中的靜態節點，並檢查 `slurm_resume.log` 文件中的動態節點以獲取錯誤詳細信息。

## 查看狀態不良的執行個體錯誤圖表

- 發生了什麼事？

已啟動數個運算執行個體，但後來終止為狀態不佳。

- 如何解決？

如需疑難排解狀態不良節點的詳細資訊，請參閱 [疑難排解非預期的節點取代和終止](#)。

## 看得見 InstanceBootstrapTimeoutError

- 發生了什麼事？

執行個體無法在 `resume_timeout` (針對動態節點) 或 `node_replacement_timeout` (針對靜態節點) 內加入叢集。如果網路未正確設定運算節點，則可能會發生這種情況，或者如果在運算節點上執行的自訂指令碼花費太長時間才能完成，則可能會發生這種情況。

- 如何解決？

對於動態節點，請檢查 `clustermgtd log (/var/log/parallelcluster/clustermgtd)` 是否有計算節點 IP 位址和錯誤，例如：

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

對於靜態節點，請檢查 `clustermgtd log (/var/log/parallelcluster/clustermgtd)` 中的計算節點 IP 位址和錯誤，如下所示：

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

如需其他詳細資訊，請檢查/var/log/cloud-init-output.log檔案是否有錯誤。您可以從clustermgtd和slurm\_resume記錄檔擷取有問題的計算節點 IP 位址。

## 看得見 EC2HealthCheckErrors

- 發生了什麼事？

執行個體無法進行 Amazon EC2 作狀態檢查。

- 如何解決？

如需如何疑難排解此問題的相關資訊，請參閱[疑難排解執行個體狀態檢查失敗](#)。

## 看得見 ScheduledEventHealthCheckErrors

- 發生了什麼事？

Amazon EC2 排定的事件運作狀態檢查執行個體失敗，而且運作狀況不佳。

- 如何解決？

如需如何疑難排解此問題的相關資訊，請參閱[執行個體的排程事件](#)。

## 看得見 NoCorrespondingInstanceErrors

- 發生了什麼事？

AWS ParallelCluster 找不到支援節點的執行個體。這些節點可能在引導操作期間自我終止。

[SlurmQueues/CustomActions/OnNodeStart| OnNodeConfigured](#)腳本，或網絡錯誤可能產生NoCorrespondingInstanceErrors。

- 如何解決？

如需其他詳細資訊，請檢查計算節點的。/var/log/cloud-init-output.log

## 查看運算叢集閒置時間圖表

### 看到明顯長於「閒置時間縮減」臨界值的 MaxDynamicNodeIdleTime

- 發生了什麼事？

您的執行個體未正確終止。MaxDynamicNodeIdleTime顯示由 Amazon EC2 執行個體支援的動態節點閒置時間上限 (以秒為單位)。「閒置時間縮減」臨界值衍生自叢集配[ScaledownIdletime](#)置參數。當計算節點閒置超過「閒置時間縮減」秒數時，請關閉節點的電Slurm源並 AWS ParallelCluster 終止支援執行個體。在這種情況下，有些東西阻止實例終止。

- 如何解決？

如需有關此問題的詳細資訊，請參閱[取代、終止或關閉有問題的執行個體和節點](#)中的[排解擴展問題](#)。

## 疑難排解叢集部署問

如果您的叢集無法建立並回復堆疊建立，您可以查看記錄檔以診斷問題。失敗訊息可能看起來像下列輸出：

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
--cluster-configuration cluster-config.yaml
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}

$ pcluster describe-cluster --cluster-name mycluster --region eu-west-1
{
  "creationTime": "2021-09-06T11:03:47.696Z",
  ...
  "cloudFormationStackStatus": "ROLLBACK_IN_PROGRESS",
  "clusterName": "mycluster",
  "computeFleetStatus": "UNKNOWN",
  "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
  "lastUpdatedTime": "2021-09-06T11:03:47.696Z",
  "region": "eu-west-1",
  "clusterStatus": "CREATE_FAILED"
}
```



## 主題

- [查看 AWS CloudFormation 事件 CREATE\\_FAILED](#)
- [使用 CLI 檢視記錄串流](#)
- [使用重新建立失敗的叢集 rollback-on-failure](#)

## 查看 AWS CloudFormation 事件 CREATE\_FAILED

您可以使用主控台或 AWS ParallelCluster CLI 檢視 CREATE\_FAILED 錯誤的 CloudFormation 事件，以協助找出根本原因。

### 主題

- [在 CloudFormation 主控台中檢視事件](#)
- [使用 CLI 來檢視和篩選 CloudFormation 事件 CREATE\\_FAILED](#)

### 在 CloudFormation 主控台中檢視事件

若要查看造成 "CREATE\_FAILED" 狀態之原因的詳細資訊，您可以使用主 CloudFormation 控台。

從主控台檢視 CloudFormation 錯誤訊息。

1. 登入 AWS Management Console 並瀏覽至 <https://console.aws.amazon.com/cloudformation>。
2. 選取名為 ## 名稱的堆疊。
3. 選擇「事件」標籤。
4. 透過按邏輯 ID 捲動資源事件清單，檢查建立失敗之資源的「狀態」。如果子任務無法創建，請向後查找失敗的資源事件。
5. 舉例來說，如果您看到下列狀態訊息，則必須使用不超過目前 vCPU 限制的執行個體類型，或要求更多 vCPU 容量。

```
2022-02-04 16:09:44 UTC-0800 HeadNode CREATE_FAILED You have requested more vCPU
capacity than your current vCPU limit of 0 allows
    for the instance bucket that the specified instance type belongs to. Please
visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to
this limit.
    (Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request
ID: a9876543-b321-c765-d432-dcba98766789; Proxy: null).
```

## 使用 CLI 來檢視和篩選 CloudFormation 事件 CREATE\_FAILED

若要診斷叢集建立問題，您可以透過篩選CREATE\_FAILED狀態來使用`pcluster get-cluster-stack-events`命令。如需詳細資訊，請參閱《使用指南》中的AWS Command Line Interface [〈篩選 AWS CLI 輸出〉](#)。

```
$ pcluster get-cluster-stack-events --cluster-name mycluster --region eu-west-1 \
  --query 'events[?resourceStatus==`CREATE_FAILED`]'
[
  {
    "eventId": "3ccdedd0-0f03-11ec-8c06-02c352fe2ef9",
    "physicalResourceId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "The following resource(s) failed to create: [HeadNode].",
  },
  {
    "eventId": "HeadNode-CREATE_FAILED-2021-09-06T11:11:51.780Z",
    "physicalResourceId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "mycluster",
    "resourceType": "AWS::CloudFormation::Stack",
    "timestamp": "2021-09-06T11:11:51.780Z"
  },
  {
    "eventId": "HeadNode-CREATE_FAILED-2021-09-06T11:11:50.127Z",
    "physicalResourceId": "i-04e91cc1f4ea796fe",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "Received FAILURE signal with UniqueId
i-04e91cc1f4ea796fe",
    "resourceProperties": "{\"LaunchTemplate\":{\"Version\":\"1\",\"LaunchTemplateId
\":\"lt-057d2b1e687f05a62\"}}",
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "HeadNode",
    "resourceType": "AWS::EC2::Instance",
    "timestamp": "2021-09-06T11:11:50.127Z"
  }
]
```

在前面的範例中，失敗是在頭節點設定中。

## 使用 CLI 檢視記錄串流

若要偵錯這類問題，您可以[pcluster list-cluster-log-streams](#)透過篩選，然後分析記錄資料流內容，來列出可從標頭節點取node-type得的記錄資料流。

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
--filters 'Name=node-type,Values=HeadNode'
{
  "logStreams": [
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      ...
    },
    ...
  ]
}
```

您可以用來尋找初始化錯誤的兩個主要記錄資料流如下：

- cfn-init是指cfn-init令碼的記錄檔。首先檢查此日誌流。您可能會在此日誌中看到Command chef failed錯誤。查看此行之前的行，以獲取與錯誤消息相關的更多細節。如需詳細資訊，請參閱 [CFN-初始化](#)。
- cloud-init是[雲端初始化](#)的記錄檔。如果您在中沒有看到任何內容cfn-init，請嘗試接下來檢查此日誌。

您可以使用 [pcluster get-cluster-log-events](#) (請注意限制擷取事件數目的--limit 5選項) 來擷取記錄資料流的內容：

```
$ pcluster get-cluster-log-events --cluster-name mycluster \  
--region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init \  
--limit 5  
{  
  "nextToken": "f/36370880979637159565202782352491087067973952362220945409/s",  
  "prevToken": "b/36370880752972385367337528725601470541902663176996585497/s",  
  "events": [  
    {  
      "message": "2021-09-06 11:11:39,049 [ERROR] Unhandled exception during build:  
Command runpostinstall failed",  
      "timestamp": "2021-09-06T11:11:39.049Z"  
    },  
    {  
      "message": "Traceback (most recent call last):\n  File \"/opt/aws/bin/  
cfn-init\", line 176, in <module>\n    worklog.build(metadata, configSets)\n  File \"/usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line  
135, in build\n    Contractor(metadata).build(configSets, self)\n  File \"/  
usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 561, in  
build\n    self.run_config(config, worklog)\n  File \"/usr/lib/python3.7/  
site-packages/cfnbootstrap/construction.py\", line 573, in run_config\n    CloudFormationCarpenter(config, self._auth_config).build(worklog)\n  File \"/usr/  
lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 273, in build\n    self._config.commands)\n  File \"/usr/lib/python3.7/site-packages/cfnbootstrap/  
command_tool.py\", line 127, in apply\n    raise ToolError(u\"Command %s failed\" %  
name)",  
      "timestamp": "2021-09-06T11:11:39.049Z"  
    },  
    {  
      "message": "cfnbootstrap.construction_errors.ToolError: Command runpostinstall  
failed",  
      "timestamp": "2021-09-06T11:11:39.049Z"  
    },  
    {  
      "message": "2021-09-06 11:11:49,212 [DEBUG] CloudFormation client initialized  
with endpoint https://cloudformation.eu-west-1.amazonaws.com",  
      "timestamp": "2021-09-06T11:11:49.212Z"  
    },  
    {  
      "message": "2021-09-06 11:11:49,213 [DEBUG] Signaling resource HeadNode in stack  
mycluster with unique ID i-04e91cc1f4ea796fe and status FAILURE",  
      "timestamp": "2021-09-06T11:11:49.213Z"  
    }  
  ]  
}
```

```
}
```

在上一個範例中，失敗是由失runpostinstall敗造成的，因此與的OnNodeConfigured組態參數中使用的自訂啟動程序指令碼的內容嚴格相關[CustomActions](#)。

## 使用重新建立失敗的叢集 **rollback-on-failure**

AWS ParallelCluster 會在 CloudWatch 記錄群組中建立叢集記錄資料流。您可以在 CloudWatch 主控台「自訂儀表板」或「記錄」群組中檢視這些記錄。如需詳細資訊，請參閱 [與 Amazon Amazon Amazon CloudWatch 的整合](#) 及 [Amazon CloudWatch 儀表](#)。如果沒有可用的記錄串流，則失敗可能是由[CustomActions](#)自訂啟動程序指令碼或 AMI 相關問題所造成。若要診斷此情況下的建立問題，請使用 [pcluster create-cluster](#) (包括設定為的--rollback-on-failure參數) 再次建立叢集false。然後，使用 SSH 檢視叢集，如下所示：

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
  --cluster-configuration cluster-config.yaml --rollback-on-failure false
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
$ pcluster ssh --cluster-name mycluster
```

登入 head 節點之後，您應該會找到三個可用來尋找錯誤的主要記錄檔。

- /var/log/cfn-init.log是指cfn-init令碼的記錄檔。首先檢查此日誌。您可能會Command chef failed在此記錄中看到錯誤訊息。查看此行之前的行，以獲取與錯誤消息相關的更多細節。如需詳細資訊，請參閱 [CFN-初始化](#)。
- /var/log/cloud-init.log是雲端初始化的記錄檔。如果您在中沒有看到任何內容cfn-init.log，請嘗試接下來檢查此日誌。
- /var/log/cloud-init-output.log是由雲初始化運行的命令的輸出。這包括來自的輸出cfn-init。在大多數情況下，您不需要查看此記錄檔即可疑難排解此類問題。

## 使用地形進行叢集部署疑難排解

本節與使用 Terraform 部署的叢集相關。

### ParallelCluster 找不到 API

規劃可能會失敗，因為找不到 ParallelCluster API。在這種情況下，返回的錯誤將是這樣的：

```
Planning failed. Terraform encountered an error while generating this plan.

#
# Error: Unable to retrieve ParallelCluster API cloudformation stack.
#
#   with provider["registry.terraform.io/aws-tf/aws-parallelcluster"],
#   on providers.tf line 6, in provider "aws-parallelcluster":
#     6: provider "aws-parallelcluster" {
#
# operation error CloudFormation: DescribeStacks, https response error StatusCode: 400,
RequestID: REQUEST_ID, api error ValidationError: Stack with id PCAPI_STACK_NAME does
not exist
```

若要解決此錯誤，請在要建立叢集的帳戶中部署 ParallelCluster API。請參閱[the section called “使用地形建立叢集”](#)。

### 未授權呼叫 ParallelCluster API 的使用者

規劃可能會失敗，因為您假設部署 Terraform 專案的 IAM 角色/使用者沒有與 API 互動的權限。ParallelCluster 在這種情況下，返回的錯誤將是這樣的：

```
Planning failed. Terraform encountered an error while generating this plan.

# Error: 403 Forbidden
#
#   with
module.parallelcluster_clusters.module.clusters[0].pcluster_cluster.managed_configs["DemoCluster"]
#   on .terraform/modules/parallelcluster_clusters/modules/clusters/main.tf line 35, in
resource "pcluster_cluster" "managed_configs":
#     35: resource "pcluster_cluster" "managed_configs" {
#
# [{"Message":"User: USER_ARN is not authorized to perform: execute-api:Invoke on
resource: PC_API_REST_RESOURCE with an explicit deny"}]
```

```
# }
```

若要解決此錯誤，請設定 ParallelCluster 提供者，使其使用 ParallelCluster API 角色與 API 互動。

```
provider "aws-parallelcluster" {  
  region      = var.region  
  profile     = var.profile  
  api_stack_name = var.api_stack_name  
  **use_user_role** **= true**  
}
```

## 排解擴展問題

本節與使用 3.0.0 版及更新 AWS ParallelCluster 版本搭配 Slurm 工作排程器安裝的叢集相關。如需配置多個佇列的詳細資訊，請參閱 [多個佇列的組態](#)。

如果其中一個執行中的叢集發生問題，請在開始疑難排解之前執行下列命令，將叢集置於 STOPPED 狀態。這樣可以防止產生任何意外的成本。

```
$ pcluster update-compute-fleet --cluster-name mycluster \  
  --status STOP_REQUESTED
```

您可以使用 [pcluster list-cluster-log-streams](#) 命令列出叢集節點中可用的日誌串流，並使用其中一個 `private-dns-name` 個失敗節點或標頭節點進行篩選：

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \  
  --filters 'Name=private-dns-name,Values=ip-10-0-0-101'
```

然後，您可以擷取記錄資料流的內容，方法是使用 [pcluster get-cluster-log-events](#) 命令並傳遞 `--log-stream-name` 對應於下一節中提到的其中一個重要記錄檔記錄檔來進行分析：

```
$ pcluster get-cluster-log-events --cluster-name mycluster \  
  --region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init
```

AWS ParallelCluster 會在 CloudWatch 記錄群組中建立叢集記錄資料流。您可以在 CloudWatch 主控台「自訂儀表板」或「記錄」群組中檢視這些記錄。如需詳細資訊，請參閱 [與 Amazon Amazon Amazon CloudWatch 的整合](#) 及 [Amazon CloudWatch 儀表](#)。

### 主題

- [調試的關鍵日誌](#)
- [slurm\\_resume.log當我無法運行作業或無法創建集群clustermgtd.log時看到InsufficientInstanceCapacity錯誤](#)
- [解決節點初始化問題](#)
- [疑難排解非預期的節點取代和終止](#)
- [取代、終止或關閉有問題的執行個體和節點](#)
- [佇列 \( 分割區 \) Inactive狀態](#)
- [疑難排解其他已知節點和工作問題](#)

## 調試的關鍵日誌

下表提供標頭節點的主要記錄檔概觀：

- `/var/log/cfn-init.log`-這是 AWS CloudFormation 初始化記錄檔。它包含了設置實例時運行的所有命令。使用它來疑難排解初始化問題。
- `/var/log/chef-client.log`-這是廚師客戶記錄 它包含了通過廚師/CINC 運行的所有命令。使用它來疑難排解初始化問題。
- `/var/log/parallelcluster/slurm_resume.log`-這是一個ResumeProgram日誌。它啟動動態節點的實例。使用它來疑難排解動態節點啟動問題。
- `/var/log/parallelcluster/slurm_suspend.log`-這是SuspendProgram記錄 當動態節點的實例終止時調用它。使用它來疑難排解動態節點終止問題。當您檢查此日誌時，您還應該檢查clustermgtd日誌。
- `/var/log/parallelcluster/clustermgtd`-這是clustermgtd記錄 它作為管理大多數群集操作操作操作的集中守護進程運行。使用它來疑難排解任何啟動、終止或叢集作業問題。
- `/var/log/slurmctld.log`-這是Slurm控制守護程序日誌。AWS ParallelCluster 不會做出縮放決策。相反，它只會嘗試啟動資源以滿足Slurm需求。它對於擴展和配置問題、工作相關問題以及任何排程器相關的啟動和終止問題都很有用。
- `/var/log/parallelcluster/compute_console_output`-此記錄會記錄來自意外終止之靜態運算節點範例子集的主控制台輸出。如果靜態運算節點終止，且中沒有計算節點記錄檔，請使用此記錄檔 CloudWatch。使用 Amazon EC2 主控台或擷取執行個體主控台輸出時，您收 AWS CLI 到的compute\_console\_output log內容相同。

以下是運算節點的主要記錄檔：



- `/var/log/cloud-init-output.log`-這是[雲端初始化](#)記錄檔。它包含了設置實例時運行的所有命令。使用它來疑難排解初始化問題。
- `/var/log/parallelcluster/computemgtd`-這是`computemgtd`記錄 它會在每個計算節點上執行，以監視主節點上`clustermgtd`守護程式離線的不常見事件中的節點。使用它來疑難排解未預期的終止問題。
- `/var/log/slurmd.log`-這是Slurm計算常駐程式記錄。使用它來疑難排解初始化和運算失敗問題。

## `slurm_resume.log`當我無法運行作業或無法創建集群 `clustermgtd.log`時看到`InsufficientInstanceCapacity`錯誤

如果叢集使用Slurm排程器，表示您遇到容量不足的問題。如果發出執行個體啟動要求時沒有足夠的可用執行個體，就會傳回`InsufficientInstanceCapacity`錯誤。

對於靜態執行個體容量，您可以在的`clustermgtd`記錄檔中找到錯誤`/var/log/parallelcluster/clustermgtd`。

對於動態執行個體容量，您可以在`ResumeProgram`記錄檔中找到錯誤，網址為`/var/log/parallelcluster/slurm_resume.log`。

訊息看起來類似下列範例：

```
An error occurred (InsufficientInstanceCapacity) when calling the RunInstances/
CreateFleet operation...
```

根據您的使用案例，請考慮使用下列其中一種方法，以避免收到這些類型的錯誤訊息：

- 停用置放群組 (如果已啟用)。如需詳細資訊，請參閱 [放置群組和執行個體啟動問題](#)。
- 保留執行個體的容量，並透過 ODCR (隨需容量保留) 啟動它們。如需詳細資訊，請參閱 [使用隨需容量保留 \(ODCR\) 啟動執行個體](#)。
- 使用不同執行個體類型設定多個運算資源。如果您的工作負載不需要特定的執行個體類型，您可以利用快速不足的容量容錯移轉與多個運算資源。如需詳細資訊，請參閱 [Slurm叢集快速容量不足容錯移轉](#)。
- 在相同的運算資源中設定多個執行個體類型，並運用多重執行個體類型配置。有關配置多個實例的更多信息，請參閱[使用 Slurm 進行多個實例類型分配和 SchedulingSlurmQueues/ComputeResources//Instances](#)。

- 在叢集配置 [Scheduling](#)/[SlurmQueues](#) 中變更子網路 ID，將佇列移至不同的可用區域 [SubnetIds](#)。 [Networking](#)
- 如果您的工作負載未緊密結合，請跨越不同的可用區域跨越佇列。如需有關設定多個子網路的詳細資訊，請參閱 [Scheduling/SlurmQueues/Networking/SubnetIds](#)。

## 解決節點初始化問題

本節說明如何疑難排解節點初始化問題。這包括節點無法啟動、開啟電源或加入叢集的問題。

### 主題

- [頭節點](#)
- [運算節點](#)

### 頭節點

適用日誌：

- `/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` 日誌或對應的日誌流。

這些記錄檔包含設定標頭節點時執行的所有動作。安裝過程中發生的大多數錯誤應該在 `/var/log/chef-client.log` 日誌中都有錯誤消息。如果在叢集的配置中指定了 `OnNodeStart` 或指 `OnNodeConfigured` 令碼，請仔細檢查指令碼是否透過記錄訊息成功執行。

建立叢集時，頭節點必須等待運算節點加入叢集，才能加入叢集。因此，如果計算節點無法加入叢集，則頭節點也會失敗。您可以依照下列其中一組程序 (視您使用的計算附註類型而定) 來疑難排解此類問題：

### 運算節點

- 適用日誌：

- `/var/log/cloud-init-output.log`
- `/var/log/slurmd.log`
- 如果啟動了計算節點，請先檢查`/var/log/cloud-init-output.log`，其中應包含類似於頭節點上日誌的安裝日誌。`/var/log/chef-client.log`安裝過程中發生的大多數錯誤應該在`/var/log/cloud-init-output.log`日誌中都有錯誤消息。如果在叢集配置中指定了安裝前或安裝後指令碼，請檢查它們是否成功執行。
- 如果您使用的是自訂 AMI 並修改Slurm組態，則可能存在Slurm相關錯誤，導致計算節點無法加入叢集。如果是排程器相關錯誤，請檢查記錄檔。`/var/log/slurmd.log`

#### 動態運算節點：

- 在 ResumeProgram log (`/var/log/parallelcluster/slurm_resume.log`) 中搜尋您的計算節點名稱，以查看ResumeProgram是否曾經使用該節點呼叫。(如果從未調用過，則可以檢查 `slurmctld log (/var/log/slurmctld.log)` 以確定ResumeProgram是否Slurm曾經嘗試ResumeProgram使用節點調用)。
- 請注意，的權限不正確ResumeProgram可能會導致ResumeProgram致無訊息失敗。如果您正在使用自定義 AMI 並進行ResumeProgram設置修改，請檢查該用slurm戶ResumeProgram是否擁有並具有744 (`rwxr--r--`) 權限。
- 如果呼ResumeProgram叫，請檢查是否已針對節點啟動執行個體。如果未啟動任何執行個體，您會看到描述啟動失敗的錯誤訊息。
- 如果執行個體已啟動，則在安裝程序期間可能發生問題。您應該會在ResumeProgram記錄檔中看到對應的私有 IP 位址和執行個體 ID。此外，您可以查看特定實例的相應設置日誌。如需有關使用計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)` 記錄檔，查看是否已針對節點啟動執行個體。如果它們沒有啟動，應該有明確的錯誤消息，詳細說明啟動失敗。
- 如果執行個體已啟動，則在安裝過程中會出現一些問題。您應該會在ResumeProgram記錄檔中看到對應的私有 IP 位址和執行個體 ID。此外，您可以查看特定實例的相應設置日誌。

#### 由 Spot 執行個體支援的運算節點：

- 如果這是您第一次使用 Spot 執行個體，且工作仍處於 PD (擱置狀態)，請仔細檢查`/var/log/parallelcluster/slurm_resume.log`檔案。您可能會發現類似以下的錯誤：

```
2022-05-20 13:06:24,796 - [slurm_plugin.common:add_instances_for_nodes] - ERROR -  
Encountered exception when launching instances for nodes (x1) ['spot-dy-t2micro-2']:  
An error occurred (AuthFailure.ServiceLinkedRoleCreationNotPermitted) when calling  
the RunInstances operation: The provided credentials do not have permission to  
create the service-linked role for Amazon EC2 Spot Instances.
```

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 AWS ParallelCluster 使用者指南 [使用 競價型執行個體](#) 中的使用者指南和 [競價型執行個體請求的服務連結角色](#)。

## 疑難排解非預期的節點取代和終止

本節將繼續探討如何疑難排解節點相關問題，特別是當節點遭到非預期取代或終止時。

- 適用日誌：
  - `/var/log/parallelcluster/clustermgtd` ( 頭節點 )
  - `/var/log/slurmctld.log` ( 頭節點 )
  - `/var/log/parallelcluster/computemgtd`(計算節點)

### 節點意外取代或終止

- 檢查 `clustermgtd log (/var/log/parallelcluster/clustermgtd)` 以查看是否 `clustermgtd` 取代或終止節點。請注意，`clustermgtd` 處理所有正常的節點維護動作。
- 如果取 `clustermgtd` 代或終止節點，應該會出現一則訊息，詳細說明為何在節點上執行此動作。如果原因與排程器相關 (例如，因為節點位於中DOWN)，請簽入 `slurmctld` 記錄以取得詳細資訊。如果原因與 Amazon EC2 相關，則應該有詳細說明需要更換的 Amazon EC2 相關問題的資訊訊息。
- 如果 `clustermgtd` 沒有終止節點，首先檢查這是否是 Amazon EC2 的預期終止，更具體地說是一個現貨終止。`computemgtd` 在計算節點上執行，如果 `clustermgtd` 判定為健康狀態不佳，也可以終止節點。檢查 `computemgtd log (/var/log/parallelcluster/computemgtd)` 以查看節點是否 `computemgtd` 終止。

## 節點失敗

- 簽入 `slurmctld log (/var/log/slurmctld.log)` 以查看工作或節點失敗的原因。請注意，如果節點失敗，工作會自動重新排入佇列。
- 如果 `slurm_resume` 報告該節點已啟動，並在幾分鐘後 `clustermgtd` 報告該節點的 Amazon EC2 中沒有對應的執行個體，則節點可能會在設定期間失敗。要從計算 ( `/var/log/cloud-init-output.log` ) 中檢索日誌，請執行以下步驟：
  - 提交工作以 Slurm 啟動新節點。
  - 等待運算節點啟動。
  - 修改執行個體起始的關機行為，以便停止失敗的計算節點而非終止。

```
$ aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --instance-initiated-shutdown-behavior "{\"Value\": \"stop\"}"
```

- 啟用終止保護。

```
$ aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --disable-api-termination
```

- 標記節點以便易於識別。

```
$ aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=Name,Value=QUARANTINED-Compute
```

- 透過變更 `parallelcluster:cluster-name` 標籤將節點從叢集中分離。

```
$ aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=parallelcluster:clustername,Value=QUARANTINED-ClusterName
```

- 使用此命令從節點擷取主控台輸出。

```
$ aws ec2 get-console-output --instance-id i-1234567890abcdef0 --output text
```

## 取代、終止或關閉有問題的執行個體和節點

- 適用日誌：
  - /var/log/parallelcluster/clustermgtd ( 頭節點 )
  - /var/log/parallelcluster/slurm\_suspend.log ( 頭節點 )
- 在大多數情況下，會clustermgtd處理所有預期的實例終止動作。簽入clustermgtd記錄檔以瞭解為何無法取代或終止節點。
- 對於動態節點失敗[SlurmSettings性質](#)，請檢查SuspendProgram日誌以查看SuspendProgram是否由特定節點作為參數調slurmctld用。請注意，實際上SuspendProgram並沒有執行任何操作。相反，它只會在調用時記錄。所有實例終止和NodeAddr重置都由完成clustermgtd。Slurm之後自動將節點置於POWER\_SAVINGsSuspendTimeout狀態。
- 如果計算節點由於啟動程序失敗而持續失敗，請驗證它們是否在啟用的情況下[Slurm叢集保護模式](#)啟動。如果未啟用受保護模式，請修改受保護模式設定以啟用受保護模式。疑難排解並修復引導腳本。

## 佇列 ( 分割區 ) Inactive狀態

如果您執行sinfo且輸出顯示AVAIL狀態為的佇列inact，表示您的叢集可能已[Slurm叢集保護模式](#)啟用，且佇列已設定為狀INACTIVE態一段預先定義的時間。

## 疑難排解其他已知節點和工作問題

另一種類型的已知問題是 AWS ParallelCluster 可能無法分配工作或做出擴展決策。針對這種類型的問題，AWS ParallelCluster 只會根據Slurm指示啟動、終止或維護資源。對於這些問題，請檢查slurmctld日誌以對其進行疑難排解。

## 放置群組和執行個體啟動問題

若要取得最低的節點間延遲，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。如果發出請求時沒有足夠的實例可用，則返回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 [SlurmQueues/Networking/PlacementGroup/Enabled](#) 參數設定為false。

若要進一步控制容量存取，請考慮使用 [ODCR \(隨需容量保留\)](#) 啟動執行個體。

[如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的執行個體啟動問題疑難排解和放置群組角色和限制。](#)

## 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家目錄 ( /home/ec2\_user在 Amazon Linux /home/centos 上 RedHat，以及 CentOS 和 Ubuntu /home/ubuntu 上 )。
- /opt/intel-這包括英特爾 MPI，英特爾並行工作室和相關文件。
- /opt/slurm-這包括Slurm工作負載管理員和相關檔案。(有條件，僅在 Scheduler: slurm 的時候。)

## 疑難排解 NICE DCV 中的問題

主題

- [NICE DCV DCV 的日誌](#)
- [NICE DCV DCV 問題](#)

### NICE DCV DCV 的日誌

NICE DCV 的記錄會寫入至/var/log/dcv/目錄中的檔案。檢閱這些記錄檔有助於疑難排解問題。

執行個體類型至少應具有 1.7 GB (GiB) 的記憶體，才能執行 NICE DCV。Nano 和微型執行個體類型沒有足夠的記憶體來執行 NICE DCV。

AWS ParallelCluster 會在記錄群組中建立 NICE DCV 記錄資料流。您可以在 CloudWatch 主控台「自訂儀表板」或「記錄」群組中檢視這些記錄。如需詳細資訊，請參閱 [與 Amazon Amazon Amazon CloudWatch 的整合](#) 及 [Amazon CloudWatch 儀表](#)。

### NICE DCV DCV 問題

在 Ubuntu 上透過 NICE 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殼層提供的 AWS ParallelCluster 使用者環境。使用者環境提供環境模組，例如 openmpi 或 intelmpi，以及其他使用者設定。

Gnome 終端機的預設設定會阻止殼層以登入殼層的形式啟動。這意味著 shell 配置文件不會自動獲取，並且不會加載 AWS ParallelCluster 用戶環境。

若要正確取得 shell 設定檔並存取 AWS ParallelCluster 使用者環境，請執行下列其中一個動作：

- 變更預設終端機設定：
  1. 在 Gnome 終端機中選擇「編輯」選單。
  2. 選取偏好設定，然後選取設定
  3. 選擇命令，然後選擇運行命令作為登錄 shell。
  4. 開啟新的終端機。
- 使用指令行來取得可用的設定檔：

```
$ source /etc/profile && source $HOME/.bashrc
```

## 使用整合進行叢 AWS Batch 集的疑難排解

本節與具有 AWS Batch 排程器整合的叢集相關。

### 主題

- [頭節點問題](#)
- [運算問題](#)
- [Job 失敗](#)
- [端點 URL 上的 Connect 逾時錯誤](#)

### 頭節點問題

您可以使用與Slurm叢集相同的方式來疑難排解主節點設定問題 (Slurm特定記錄除外)。如需有關這些問題的詳細資訊，請參閱[頭節點](#)。

### 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 AWS Batch [疑難排解](#) 說明文件以取得協助。

### Job 失敗

如果工作失敗，您可以執行[awsbcout](#)命令來擷取工作輸出。您也可以執行[awsbstat](#)命令以取得 Amazon 儲存之任務日誌的連結 CloudWatch。



## 端點 URL 上的 Connect 逾時錯誤

如果多節點 parallel 作業失敗並出現錯誤：Connect timeout on endpoint URL

- 在awsbout輸出日誌中，檢查工作是否與輸出 parallel 多節點：Detected 3/3 compute nodes. Waiting for all compute nodes to start.
- 驗證計算節點子網路是否為公用。

在中使用時，多節點 parallel 作業不支援使 AWS Batch 用公用子網路。AWS ParallelCluster針對您的運算節點和工作使用私有子網路。如需詳細資訊，請參閱《AWS Batch 使用指南》中的[計算環境考量](#)。若要為您的計算節點設定私有子網路，請參閱[AWS ParallelCluster 與 AWS Batch 調度程序](#)。

## 疑難排解多使用者與活動目錄整合

本節與使用中目錄整合的叢集有關。

如果使用中目錄整合功能無法如預期般運作，SSSD 記錄檔可以提供有用的診斷資訊。這些記錄位於叢集節點/var/log/sss上。依預設，它們也會儲存在叢集的 Amazon CloudWatch 日誌群組中。

### 主題

- [活動目錄特定的故障](#)
- [啟用除錯模式](#)
- [如何從網路轉移到 LDAP](#)
- [如何停用 LDAPS 伺服器憑證驗證](#)
- [如何使用 SSH 密鑰而不是密碼登錄](#)
- [如何重置用戶密碼和過期的密碼](#)
- [如何驗證加入的網域](#)
- [如何疑難排解憑證問題](#)
- [如何驗證與活動目錄的集成是否正常工作](#)
- [如何疑難排解登入運算節點的問題](#)
- [在多使用者環境中，SimCenter StarCCM+ 工作的已知問題](#)
- [用戶名解析的已知問題](#)
- [如何解決主目錄創建問題](#)

## 活動目錄特定的故障

本節與特定於「活動目錄」類型的疑難排解相關。

### Simple AD

- 此DomainReadOnlyUser值必須符合使用者的 Simple AD 目錄庫搜尋：

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

注cn意Users。

- 預設的管理員使用者為Administrator。
- Ldapsearch需要在使用者名稱之前使用 NetBIOS 名稱。

Ldapsearch語法必須如下所示：

```
$ ldapsearch -x -D "corp\\Administrator" -w "Password" -H ldap://192.0.2.103 \  
-b "cn=Users,dc=corp,dc=example,dc=com"
```

### AWS Managed Microsoft AD

- 此DomainReadOnlyUser值必須符合使用者的 AWS Managed Microsoft AD 目錄庫搜尋：

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

- 預設的管理員使用者為Admin。
- Ldapsearch語法必須如下所示：

```
$ ldapsearch -x -D "Admin" -w "Password" -H ldap://192.0.2.103 \  
-b "ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

## 啟用除錯模式

SSSD 的偵錯記錄檔對於疑難排解問題很有用。若要啟用偵錯模式，您必須使用對叢集配置所做的下列變更來更新叢集：

```
DirectoryService:  
  AdditionalSssdConfigs:
```

```
debug_level: "0x1fff"
```

## 如何從網路轉移到 LDAP

不建議從 LDAPS (含有 TLS/SSL 的 LDAP) 移至 LDAP，因為單獨使用 LDAP 不提供任何加密功能。但是，它可以用於測試目的和故障排除。

您可以使用先前的配置定義更新叢集，將叢集還原為先前的配置。

若要從 LDAPS 移至 LDAP，您必須使用叢集配置中的下列變更來更新叢集：

```
DirectoryService:
  LdapTlsReqCert: never
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

## 如何停用 LDAPS 伺服器憑證驗證

為了測試或疑難排解，暫時停用頭節點上的 LDAPS 伺服器憑證驗證會很有用。

您可以使用先前的配置定義更新叢集，將叢集還原為先前的配置。

若要停用 LDAPS 伺服器憑證驗證，您必須使用叢集配置中的下列變更來更新叢集：

```
DirectoryService:
  LdapTlsReqCert: never
```

## 如何使用 SSH 密鑰而不是密碼登錄

SSH 金鑰會在您第一 `/home/$user/.ssh/id_rsa` 次使用密碼登入之後建立。要使用 SSH 密鑰登錄，您必須使用密碼登錄，在本地複製 SSH 密鑰，然後像往常一樣使用它來使用 SSH 密碼：

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY username@$head_node_ip
```

## 如何重置用戶密碼和過期的密碼

如果使用者無法存取叢集，則其 [AWS Managed Microsoft AD 密碼可能已過期](#)。

若要重設密碼，請以具有目錄寫入權限的使用者和角色執行下列命令：

```
$ aws ds reset-user-password \  
  --directory-id "d-abcdef01234567890" \  
  --user-name "USER_NAME" \  
  --new-password "NEW_PASSWORD" \  
  --region "region-id"
```

如果您重設 [DirectoryService](#) 的密碼 [DomainReadOnlyUser](#) :

1. 請務必使用新密碼更新 [DirectoryService/PasswordSecretArn](#) 秘密。
2. 更新新密碼值的叢集：
  - a. 使用 `pcluster update-compute-fleet` 指令停止運算叢集。
  - b. 從叢集頭節點內執行下列命令。

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

在密碼重設和叢集更新之後，應該還原使用者的叢集存取權。

如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的「[重設使用者密碼](#)」。

## 如何驗證加入的網域

下列命令必須從加入網域的執行個體執行，而不是頭節點執行。

```
$ realm list corp.example.com \  
type: kerberos \  
realm-name: CORP.EXAMPLE.COM \  
domain-name: corp.example.com \  
configured: kerberos-member \  
server-software: active-directory \  
client-software: sssd \  
required-package: oddjob \  
required-package: oddjob-mkhomedir \  
required-package: sssd \  
required-package: adcli \  
required-package: samba-common-tools \  
login-formats: %U \  
login-policy: allow-realm-logins
```

## 如何疑難排解憑證問題

當 LDAPS 通訊不起作用時，可能是由於 TLS 通訊中的錯誤，這反過來可能是由於憑證問題所致。

憑證相關注意事項：

- 叢集設定中指定的憑證 `LdapTlsCaCert` 必須是 PEM 憑證組合，其中包含針對網域控制站核發憑證之整個授權憑證 (CA) 鏈結的憑證。
- PEM 憑證組合包是由 PEM 憑證串連所組成的檔案。
- PEM 格式的憑證 (通常在 Linux 中使用) 等同於 Base64 DER 格式的憑證 (通常由視窗匯出)。
- 如果網域控制站的憑證是由下屬 CA 發行，則憑證套裝軟體必須同時包含從屬 CA 和根 CA 的憑證。

疑難排解驗證步驟：

下列驗證步驟假設命令是從叢集標頭節點內執行，而且網域控制站可在 `SERVER:PORT`。

若要疑難排解與憑證相關的問題，請依照下列驗證步驟執行：

驗證步驟：

1. 檢查到活動目錄域控制站的連接：

確認您可以連線到網域控制站。如果此步驟成功，則與網域控制站的 SSL 連線會成功，並驗證憑證。您的問題與憑證無關。

如果此步驟失敗，請繼續進行下一次驗證。

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

2. 檢查憑證驗證：

確認本機 CA 憑證服務包可驗證網域控制站所提供的憑證。如果此步驟成功，則您的問題與憑證無關，而是與其他網路問題有關。

如果此步驟失敗，請繼續進行下一次驗證。

```
$ openssl verify -verbose -  
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

3. 檢查活動目錄域控制站提供的證書：

確認網域控制站提供的憑證內容如預期般。如果這個步驟成功，您可能有用來驗證控制器的 CA 憑證的問題，請移至下一個疑難排解步驟。

如果這個步驟失敗，您必須更正為網域控制站發行的憑證，並重新執行疑難排解步驟。

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

#### 4. 檢查證書的內容：

確認網域控制站所提供的憑證內容如預期般。如果此步驟成功，您可能遇到用於驗證控制器的 CA 憑證的問題，請移至下一個疑難排解步驟。

如果這個步驟失敗，您必須更正為網域控制站發行的憑證，並重新執行疑難排解步驟。

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

#### 5. 檢查本機 CA 憑證套裝軟體的內容：

確認用來驗證網域控制站憑證之本機 CA 憑證套裝軟體的內容如預期般。如果這個步驟成功，您可能會遇到網域控制站所提供的憑證的問題。

如果此步驟失敗，您必須更正針對網域控制站發行的 CA 憑證套件，並重新執行疑難排解步驟。

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

## 如何驗證與活動目錄的集成是否正常工作

如果下列兩項檢查成功，則與作用中目錄的整合正在運作。

檢查：

#### 1. 您可以探索目錄中定義的使用者：

從叢集頭節點內，如下所示 `ec2-user`：

```
$ getent passwd $ANY_AD_USER
```

#### 2. 您可以使用 SSH 進入提供使用者密碼的頭節點：

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

如果檢查一個失敗，我們預計檢查兩個也失敗。

其他疑難排解檢查：

- 確認使用者存在於目錄中。
- 啟動 [偵錯記錄](#)。
- 請考慮暫時停用加密，方法是 [從 LDAPS 移至 LDAP](#)，以排除 LDAPS 問題。

## 如何疑難排解登入運算節點的問題

本節與登入到與 Active Directory 整合的叢集中的計算節點相關。

使用時 AWS ParallelCluster，叢集運算節點的密碼登入會停用設計。

所有使用者都必須使用自己的 SSH 金鑰來登入計算節點。

如果在叢集配置中啟用，使用者可以在第一次驗證後 (例如 [GenerateSshKeysForUsers](#) 登入) 在 head 節點中擷取其 SSH 金鑰。

當使用者第一次在頭節點上進行驗證時，可擷取以目錄使用者身分自動產生的 SSH 金鑰。也會建立使用者的主目錄。這也可能發生在 sudo-user 第一次切換到頭節點中的用戶時。

如果使用者尚未登入頭節點，則不會產生 SSH 金鑰，且使用者將無法登入計算節點。

## 在多使用者環境中，SimCenter StarCCM+ 工作的已知問題

本節與西門子 Simcenter StarCCM+ 計算流體動力學軟件在多用戶環境中推出的工作相關。

如果您執行設定為使用內嵌式 Intel MPI 的 StarCCM+ v16 工作，則預設情況下，MPI 程序會使用 SSH 啟動載入。

由於導致用戶名解析錯誤 [Slurm](#) 的已知錯誤，作業可能會失敗並顯示類似的錯誤 error setting up the bootstrap proxies。此錯誤僅影響 AWS ParallelCluster 版本 3.1.1 和 3.1.2。

若要避免這種情況發生，請強制 Intel MPI Slurm 作為 MPI 啟動程序方法使用。如 [IntelMPI](#) 官方文件中所述，`I_MPI_HYDRA_BOOTSTRAP=slurm` 將環境變數匯出至啟動 StarCCM+ 的工作程序檔。

## 用戶名解析的已知問題

本節與檢索作業中的用戶名有關。

由於 [Slurm 中存在已知錯誤](#)，在作業程序中擷取的使用者名稱可能是nobody如果您沒有執行作業。srun此錯誤僅影響 AWS ParallelCluster 版本 3.1.1 和 3.1.2。

例如，如果您以目錄使用者sbatch --wrap 'srun id'身分執行命令，則會傳回正確的使用者名稱。不過，如果您以目錄使用者sbatch --wrap 'id'身分執行，則nobody可能會以使用者名稱的形式傳回。

您可以使用下列因應措施。

1. 如果可能的話 'sbatch'，請使用 'srun' 而不是啟動您的工作。
2. 透過設定叢集[AdditionalSssdConfigs](#)中的配置來啟用 SSSD 列舉，如下所示。

```
AdditionalSssdConfigs:  
  enumerate: true
```

## 如何解決主目錄創建問題

本節與建立主目錄相關的問題。

如果您看到類似下列範例所示的錯誤，則當您第一次登入 head 節點時，並不會為您建立主目錄。或者，當您第一次從 sudoer 切換到頭節點中的 Active Directory 使用者時，不會為您建立主目錄。

```
$ ssh AD_USER@$HEAD_NODE_IP  
/opt/parallelcluster/scripts/generate_ssh_key.sh failed: exit code 1  
  
  _|  _|_ )  
 _| (    /  Amazon Linux 2 AMI  
__|\___|__|  
  
https://aws.amazon.com/amazon-linux-2/  
Could not chdir to home directory /home/PclusterUser85: No such file or directory
```

主目錄建立失敗可能是因為叢集頭節點中安裝的oddjob和oddjob-mkhomedir套件所造成。

如果沒有主目錄和 SSH 金鑰，使用者就無法將作業或 SSH 提交至叢集節點。



如果您的系統中需要oddjob套件，請確認oddjobd服務正在執行，並重新整理 PAM 組態檔案，以確定已建立主目錄。若要執行這項操作，請在 head 節點中執行指令，如下列範例所示。

```
sudo systemctl start oddjobd
sudo authconfig --enablemkhomedir --updateall
```

如果您不需要系統中的oddjob套件，請解除安裝套件並重新整理 PAM 設定檔，以確定已建立主目錄。若要執行這項操作，請在 head 節點中執行指令，如下列範例所示。

```
sudo yum remove -y oddjob oddjob-mkhomedir
sudo authconfig --enablemkhomedir --updateall
```

## 排解自訂 AMI 問題

當您使用自訂 AMI 時，您可以看到下列警告：

```
"validationMessages": [
  {
    "level": "WARNING",
    "type": "CustomAmiTagValidator",
    "message": "The custom AMI may not have been created by pcluster. You can ignore this warning if the AMI is shared or copied from another pcluster AMI. If the AMI is indeed not created by pcluster, cluster creation will fail. If the cluster creation fails, please go to https://docs.aws.amazon.com/parallelcluster/latest/ug/troubleshooting.html#troubleshooting-stack-creation-failures for troubleshooting."
  },
  {
    "level": "WARNING",
    "type": "AmiOsCompatibleValidator",
    "message": "Could not check node AMI ami-0000012345 OS and cluster OS alinux2 compatibility, please make sure they are compatible before cluster creation and update operations."
  }
]
```

如果您確定正在使用正確的 AMI，則可以忽略這些警告。

如果您 future 不想看到這些警告，請使用以下標籤標記自訂 AMI，其中 *my-os* 是 alinux2、ubuntu2204、或之一 ubuntu2004centos7，rhel8 而 *#3.7.0#* 是使用中的 pcluster 版本：

```
$ aws ec2 create-tags \  
  --resources ami-yourcustomAmi \  
  --tags Key="parallelcluster:version",Value="3.7.0" \  
  Key="parallelcluster:os",Value="my-os"
```

## 疑難排解cfn-hup未執行時的叢集更新逾時

cfn-hupHelper 是一種精靈，可偵測資源中繼資料的變更，並在偵測到變更時執行使用者指定的動作。這是您透過 UpdateStack API 動作在執行中的 Amazon EC2 執行個體上進行組態更新的方式。

目前cfn-hup守護程式是由supervisord。但是啟動後，該cfn-hup過程將脫離supervisord控制。如果cfn-hup惡魔被外部演員殺死，它不會自動重新啟動。如果cfn-hup未執行，則在叢集更新期間，CloudFormation 堆疊會如預期般啟動更新程序，但不會在頭節點上啟動更新程序，且堆疊最終會進入逾時狀態。從群集日誌中/var/log/chef-client，您可以看到從未調用更新方案。

檢查並cfn-hup在發生故障時重新啟動

1. 在頭節點上，檢查cfn-hup是否正在運行：

```
$ ps aux | grep cfn-hup
```

2. 檢查cfn-hup日誌/var/log/cfn-hup.log和頭節點/var/log/supervisord.log上。
3. 如果cfn-hup沒有運行，請嘗試通過運行重新啟動它：

```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl \  
start cfn-hup
```

## 網路疑難排

### 單一公用子網路中的叢集問題

cloud-init-output.log從其中一個計算節點檢查。如果您發現類似下列項目，表示節點在Slurm初始化中卡住，很可能是因為遺失 DynamoDB VPC 端點所致。新 DynamoDB 點。如需更多資訊，請參閱[AWS ParallelCluster 在沒有互聯網訪問的單個子網中](#)。

```
ruby_block[retrieve compute node info] action run[2022-03-11T17:47:11+00:00] INFO: \  
Processing ruby_block[retrieve compute node info] action run (aws-parallelcluster- \  
slurm::init line 31)
```

## onNodeUpdated自訂動作的叢集更新失敗

當 [HeadNode/CustomActions/OnNodeUpdated](#) 腳本失敗時，更新失敗，腳本不會在回滾時運行。您有責任在復原完成之後手動執行所需的清除作業。例如，如果OnNodeUpdated指令碼變更組態檔中某個欄位的狀態 (例如從true到false)，然後失敗，您就必須手動將該欄位值還原為更新前的狀態 (例如，false到true)。如需詳細資訊，請參閱 [自訂引導動作](#)。

## 查看自定義配Slurm置的錯誤

從 AWS ParallelCluster 版本 3.6.0 開始，您不能再將單個prolog或epilog腳本包含在自定義Slurm配置中來定位它們。在 AWS ParallelCluster 版本 3.6.0 及更高版本中，您必須在相應的資料夾prolog和Epilog資料夾中找到自訂Prolog和epilog指令碼。依預設，這些資料夾設定為指向：

- Prolog指向/opt/slurm/etc/scripts/prolog.d/。
- Epilog指向/opt/slurm/etc/scripts/epilog.d/。

我們建議您保留90\_plcluster\_health\_check\_manager序言腳本和 90\_pcluster\_noop epilog腳本到位。

Slurm以相反的字母順序執行指令碼。Prolog和資Epilog料夾必須至少包含一個檔案。如需詳細資訊，請參閱 [Slurmprolog 與 epilog](#) 及 [Slurm組態自訂](#)。

## 叢集警示

叢集健康狀態監控對於確保最佳效能至關重要。AWS ParallelCluster 可讓您監視叢集頭節點的多個CloudWatch 警示。

本節提供每種類型的「頭節點叢集警示」的詳細資訊，包括其命名慣例、觸發警示的特定條件，以及建議的疑難排解步驟。

叢集警示的命名慣例為CLUSTER\_NAME-COMPONENT-METRIC，例如mycluster-HeadNode-Cpu。

- CLUSTER\_NAME-HeadNode: 信號頭節點的整體狀態。如果以下至少有一個警報是紅色的。
- CLUSTER\_NAME-HeadNode-Health：如果至少有一個 Amazon EC2 運作 Health 檢查失敗，則為紅色。如果發生警報，我們建議您查看[使用失敗狀態檢查進行故障排除的實例](#)。
- CLUSTER\_NAME-HeadNode-Cpu：如果 CPU 使用率大於 90%，則為紅色。如果發生警報，請檢查消耗最多 CPU 的進程 `ps -aux --sort=-%cpu | head -n 10`。

- CLUSTER\_NAME-HeadNode-Mem：如果記憶體使用率大於 90%，則為紅色。在發生警報的情況下，請檢查消耗最多內存的進程 `ps -aux --sort=-%mem | head -n 10`。
- CLUSTER\_NAME-HeadNode-Disk：如果路徑上的佔用磁盤空間大於 90%，則為紅色。如果發生警報，請檢查佔用大部分空間的文件夾 `du -h --max-depth=2 / 2> /dev/null | sort -hr`。

## 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。

對於更緊急的問題，請聯繫 AWS Support 或打開一個 [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 Support 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 |
|------------------------|--------------------|
| 3.0. x                 | 3/31/2023          |
| 3.1. x                 | 8/31/2023          |
| 3.2. x                 | 1/31/2024          |
| 3.3. x                 | 5/31/2024          |
| 3.4. x                 | 6/28/2024          |
| 3.5. x                 | 8/31/2024          |
| 3.6. x                 | 11/30/2024         |
| 3.7. x                 | 2025 年 2 月 28 日    |
| 3.8. x                 | 2025 年 6 月 30 日    |
| 3.9. x                 | 2025 年 5 月 9 日     |
| 3.10. x                | 2025 年 12 月 27 日   |

# 中的安全性 AWS ParallelCluster

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您之間共同承擔的責任。[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：

- 雲端的安全性 — AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎架構。AWS 還為您提供可以安全使用的服務。若要深入瞭解適用於的規範遵循計劃 AWS ParallelCluster，請參閱[合規計劃的 AWS 服務範圍內的](#)。
- 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。下列主題說明如何設定 AWS ParallelCluster 以符合安全性與合規性目標。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

- [Amazon EC2 中的安全性](#)
- [Amazon API Gateway 中的安全性](#)
- [中的安全性 AWS Batch](#)
- [AWS CloudFormation 中的安全性](#)
- [Amazon 的安全 CloudWatch](#)
- [AWS CodeBuild 中的安全性](#)
- [亞馬遜動態 B 中的安全](#)
- [Amazon ECR 中的安全性](#)

- [Amazon ECS 中的安全](#)
- [Amazon EFS 中的安全](#)
- [FSx 中針對光澤的安全性](#)
- [AWS Identity and Access Management \(IAM\) 中的安全性](#)
- [EC2 Image Builder 中的安全性](#)
- [中的安全性 AWS Lambda](#)
- [Amazon 路線 53 中的安全](#)
- [Amazon SNS 中的安全性](#)
- [Amazon SQS 中的安全性 \(適用於 AWS ParallelCluster 版本 2.x。\)](#)
- [Amazon S3 中的安全性](#)
- [Amazon VPC 中的安全性](#)

## 資料保護 AWS ParallelCluster

AWS [共用責任模型](#)適用於中的資料保護 AWS ParallelCluster。如此模型中所述，AWS 負責保護執行所有 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務的安全組態和管理任務。如需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 安全性GDPR部落格上的[AWS 共同責任模型和部落格文章](#)。

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

- 對每個帳戶使用多重要素驗證 (MFA)。
- 使用SSL/TLS與 AWS 資源溝通。我們需要 TLS 1.2 並推薦 TLS 1.3。
- 使用設定API和使用者活動記錄 AWS CloudTrail。
- 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 AWS 透過命令列介面或存取時需要 FIPS 140-3 驗證的密碼編譯模組API，請使用端點。FIPS 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。這包括當您與控制台、API、AWS ParallelCluster 或一起 AWS 服務 使用或使用其他控制台時 AWS SDKs。AWS CLI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 除了代表使用者與 AWS 服務互動所需的憑證外，本身不會儲存任何客戶資料。

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，使用區 [EbsSettings](#) 段中的 [EbsSettings/Encrypted](#) 和 [EbsSettings/KmsKeyId](#) 設定來設定加密。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，使用區 [EfsSettings](#) 段中的 [EfsSettings/Encrypted](#) 和 [EfsSettings/KmsKeyId](#) 設定來設定加密。[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 來叫用將客戶資料傳輸到本機電腦以進行儲存的 AWS 服務，請參閱該服務使用者指南中的「安全性與合規性」一章，以取得有關如何儲存、保護和加密該資料的資訊。

### 傳輸中加密

根據預設，從執行中的用戶端電腦 AWS ParallelCluster 和 AWS 服務端點傳輸的所有資料都會透過 HTTPS/TLS連線傳送所有資料來加密。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 權限 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/ssr/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。

如需特定合規計畫的 AWS 服務範圍清單，請參閱[合規計畫的 AWS 服務範圍](#)。如需一般資訊，請參閱[AWS 合規計畫](#)。

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。
- [Amazon Web Services 的 HIPAA 安全與合規架構白皮書 — 本AWS白皮書](#)說明公司可如何運用 AWS 來建立符合 HIPAA 規範的應用程式。
- [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

為提高與AWS服務通訊時的安全性，您應該設定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。

為確保不AWS ParallelCluster使用 TLS 1.2 之前的 TLS 版本，您可能必須重新編譯 OpenSSL 來強制執行此最低版本限制，然後重新編譯 Python 以使用新建立的 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。

# 版本說明和文件記錄

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         |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <a href="#">AWS ParallelCluster 推出使用者介面版本</a> | <p>我們很高興宣布推出 AWS ParallelCluster 使用者介面版本 2024.07.1。</p> <p>變更：</p> <ul style="list-style-type: none"><li>• 添加對 AWS ParallelCluster 3.10.1 的支持。</li></ul> <p>錯誤修正：</p> <ul style="list-style-type: none"><li>• 修正了打破工作會計信息渲染的錯誤。</li><li>• 修復了功能標記機制中的一個錯誤，該機制禁用 PC 3.10.0+ 上的所有 PC 3.2.0+ 功能。</li></ul> <p>安全性：</p> <ul style="list-style-type: none"><li>• 將密碼編譯從 42.0.2 升級到 42.0.4，並指定JWT解決弱點和 <a href="https://cwe.mitre.org/data/definitions/476.html">https://cwe.mitre.org/data/definitions/476.html</a><a href="https://cwe.mitre.org/data/definitions/327.html">https://cwe.mitre.org/data/definitions/327.html</a></li><li>• 將 Node.js 版本從 13.5.1 升級至 14.1.1 版本，以解決這些漏洞。 <a href="https://cwe.mitre">https://cwe.mitre</a></li></ul> | 2024年7月24日 |

[.org/data/definitions/91](https://cwe.mitre.org/data/definitions/918.html)

[8.html](https://cwe.mitre.org/data/definitions/918.html)Next.js

- 將 ws 版本限制為 8.17.1+ 以上，以解決漏洞。<https://cwe.mitre.org/data/definitions/476.html>

查看 [完整的更新日誌](#)。

### [AWS ParallelCluster 版本:版本發布](#)

我們很高興地宣布 AWS ParallelCluster 3.10.1 的發布。

2024年7月8日

錯誤修正：

- 修復中國地區的圖像構建失敗。

### [AWS ParallelCluster 推出使用者介面版本](#)

我們很高興宣布推出 AWS ParallelCluster 使用者介面版本 2024.07.0。

2024年7月2日

功能：

- 增加了對 3.10.0 AWS ParallelCluster 版本的支持。

## [AWS ParallelCluster 版本](#)

### [3.10.0 已發布](#)

我們很高興地宣布 AWS ParallelCluster 3.10.0 的發布

2024年6月27日

若要升級，請鍵入 `sudo pip install --upgrade aws-parallelcluster` 。

增強功能：

- 添加新的配置部分 `Scheduling/SlurmSettings/ExternalSlurmdbd` 以將集群連接到外部 `Slurmdbd`。
- 允許在隔離的網絡中運行構建映像。
- 添加對 Amazon 2023 的支持。
- 將支援新增 `price-capacity-optimized` 為 `AllocationStrategy` 。
- 新增驗證程式，以防止將置入群組與容量區塊搭配使用。

變更：

- 已不再支援 CentOS 7。
- 將 CINC 用戶端從 18.2.7 升級至 18.4.12 的版本。
- 升 `munge` 到 0.5.16 版本 ( 從 0.5.15 )。
- 將混音器升級到 5.0.2 ( 從 4.2.9 )。



- 升級第三方食譜依賴關係：
  - 四月七日至五月二十二日  
(由四月七日起)
  - 開放時間 -2.11.12 (從開放時間 -2.11.3)
- 刪除第三方食譜:半導體  
-6.1.12.
- 將EFA安裝程式升級至1.32.0.
  - EFA-驅動程序：  
efa-2.8.0-1
  - EFA-配置：efa-config-1.16-1
  - 外匯基金會概況：efa-profile-1.7-1
  - 利布-AWS：libfabric-aws-1.21.0-1
  - RDMA 核心：rdma-core-50.0-1
  - 打開MPI：openmpi40-aws-4.1.6-3 和 openmpi50-aws-5.0.2-12
- 將NVIDIA驅動程式升級至版本 535.183.01 (從 5 年 5 月 5 日起)。
- 將 Python 升級到 3.9.19 (從 3.9.17 開始)。
- 將英特爾MPI庫升級到 2021.12.1.8 (從 2021.9.0.43482)。

錯誤修正：

- 修復數據存儲庫關聯配置以製作AutoExportPolicy 和AutoImportPolicy 可選。
- 修正叢集刪除期間的問題，此問題現在會在執行個體處於往返或終止狀態時完成運算叢集清理。(這是為了避免終止週期較長的執行個體類型發生叢集刪除失敗。
- 允許在叢集配置的Monitoring 區段中啟用cloudwatch 儀表板並停用警示。
- 允許 ParallelCluster 自訂資源使用隱藏驗證程式。PclusterCluster/ SuppressValidators
- 刪除，/etc/profile.d/pcluster.sh 以便它不會在每個用戶登錄時執行，並且cfn\_bootstrap\_virtualenv 不會添加到PATH 環境變量中。
- 通過用DescribeCluster 響應替換字段failureReason 來failures修復ParallelCluster API 規範。
- 透過新增遺失的 CloudFormation 堆疊狀態來修正ParallelCluster API 規格:IMPORT\_\*REVIEW\_IN

\_PROGRESS 、  
和UPDATE\_FAILED 。

- 修正叢集更新無法在傳輸過程中包含具有加密功能的EFS檔案系統的問題。
- 修正當用於共用內部資料時，slurmctld 和 slurmdbd 服務無法在頭節點重新開機時EFS重新啟動的問題。
- 在 Ubuntu 系統上，移除與來自平行叢集的配置衝突的雲端初始化記錄檔的預設 logrotate 設定。
- 使用 RHEL 8.10 或更高版本修復映像構建失敗。

### [1.0.0 版的地形表單提供者發 AWS ParallelCluster 布](#)

我們很高興宣布推出 1.0.0 版的地形表單提供 AWS ParallelCluster 者。

2024年6月26日

功能：

- [完整的更新日誌](#)

### [1.0.0 版的地形模塊發 AWS ParallelCluster 布](#)

我們很高興宣布推出 1.0.0 版的地形模組。AWS ParallelCluster

2024年6月26日

功能：

- [完整的更新日誌](#)

## [AWS ParallelCluster 版本](#)

### [3.9.3 發布](#)

我們很高興地宣布 AWS ParallelCluster 3.9.3 的發布

2024年6月19 日

若要升級，請輸入 `sudo pip install --upgrade aws-parallelcluster`

功能：

- 已新增對中FSx Lustre做為共用儲存類型的支援us-iso-east-1。

錯誤修正：

- `cloud_dns` 從 Slurm 配置 `SlurmctldParameters` 中刪除以避免 Slurm 扇出問題。

這不是必要的，因為我們會在執行個體啟動時設定 IP 位址。

## [AWS ParallelCluster 版本](#)

### [3.9.2 發布](#)

我們很高興地宣布 AWS ParallelCluster 3.9.2 的發布

2024年5月28日

功能：

- 升級Slurm至十一點七（從11 點開始）
- 有關更多內容，敬請參閱（詳CHANGELOG [3.9.2](#)見）GitHub。

### [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。

2024 年 5 月 14 日

錯誤修正：

- 修復了當用戶打開「Job 狀態」面板時，前端會阻止用戶界面的錯誤。
- [完整的更新日誌](#)

### [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 已發佈使用者介面版本。

2024年4月17日

功能：

- 增加了對 3.9.1 AWS ParallelCluster 版本的支持
- [完整的更新日誌](#)

### [AWS ParallelCluster 版本 3.9.1 版本已發布](#)

我們很高興地宣布 AWS ParallelCluster 3.9.1 的發布

2024年4月11日

若要升級，請輸入下列內容：  
`sudo pip install --upgrade aws-parallelcluster`

錯誤修正

- 在更新叢集作業中卸載檔案系統時，移除共用儲存裝置 mountdir 的遞迴刪除。

## [AWS ParallelCluster 版本 3.9.1 版本已發布](#)

我們很高興地宣布 AWS  
ParallelCluster 3.9.1 的發布

2024年4月11日

若要升級，請輸入下列內  
容：`sudo pip install  
--upgrade aws-parallelcluster`

錯誤修正

- 在更新叢集作業中卸載檔案系統時，移除共用儲存裝置 mountdir 的遞迴刪除。

## [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。

2024年3月12日

功能：

- 增加了對 3.9.0 AWS ParallelCluster 版本的支持
- 增加了對 Ubuntu 22.04 和紅帽企業 Linux 9 的支持
- 不推薦使用的 Ubuntu

錯誤修正

- 修正使用多個叢集時，部分叢集無法顯示的問題

有關變更的詳細資訊，請參閱（詳見）的[aws-parallelcluster-ui](#)套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 版本](#)

### [3.9.0 已發布](#)

我們很高興地宣布 AWS ParallelCluster 3.9.0 的發布

2024年3月5日

若要升級，請輸入下列內容：  
`sudo pip install --upgrade aws-parallelcluster`

增強功能：

- 新增組態參數，DeploymentSettings/DefaultUserHome 允許使用者將預設使用者的主目錄移至，/local/home 而非 /home (預設值)。
- 允許更新 MinCountMaxCount、Queue和ComputeResource 設定參數，而不需要停止運算叢集。現在可以通過設置為更新它Scheduling/SlurmSettings/QueueUpdateStrategy 們TERMINATE。AWS ParallelCluster 只會終止透過叢集更新執行的叢集容量調整大小期間移除的節點。
- 允許更新 EFS、FSxLustre、FSxOntap、FSxOpenZfs 且 FileCache 不取代運算和登入叢集。
- 添加支持RHEL9.
- 通過過build-image 程CustomAmi 創建

添加對洛基 Linux 9 的支持。目前沒有公共官方的 AWS ParallelCluster Rocky9 Linux AMI 可用。

- CommunicationParameters 從 Slurm 「自訂設定」拒絕清單中移除。
- 添加DeploymentSettings/DisableSudoAccessForDefaultUser 參數以禁用支持OSes的默認用戶的 sudo 訪問。
- FSx對於 Lustre 檔案系統所建立者的變更 ParallelCluster：將 Lustre 伺服器版本變更為 2.15。
- 通過['cluster']['nvidia']['kernel\_open'] 食譜節點屬性構建時AMI，添加在開源和封閉源 Nvidia 驅動程序之間進行選擇的可能性。
- \* 新增 clustermgtd 組態選項，ec2\_instance\_missing\_max\_count 以允許最終 Amazon 的可設定重試次數，EC2描述執行個體與執行個體的一致性。

## 變更

- 升級Slurm至二點一點一（從2月7日起）



- 將NVIDIA驅動程式升級至版本
- 在集群中添加對 Python 3.11, 3.12 的支持。CLI `aws-parallelcluster-batch-cli`
- 使用 Amazon EC2 DescribeInstances 回應NetworkCardIndex 清單中的網路卡索引來建立網路界面，而不是在MaximumNetworkCards 範圍內迴圈。
- 使用執行個體類型 P3、G3、P2 和 G2 時，建立叢集失敗，因為它們的 GPU架構與 3.8.0 版本中引入的開放原始碼 Nvidia 驅動程式 (OpenRM) 不相容。
- 升級第三方食譜相依性：`nfs-5.1.2` (從 `nfs-5.0.0` 起)
- 升級EFA安裝程式至 `1.30.0`.
  - EFA-驅動程序：`efa-2.6.0-1`
  - EFA-配置：`efa-config-1.15-1`
  - 外匯基金會概況：`efa-profile-1.6-1`
  - 利布-AWS：`libfabric-aws-1.19.0`
  - RDMA 核心：`rdma-core-46.0-1`
  - 打開MPI：`openmpi40-aws-4.1.6-2` 和

openmpi50-aws-5.0.  
0-11

- 升級NICEDCV至版本  
2023.1-16388.
  - 伺服器：2023.1.16  
388-1
  - xdcv：2023.1.565-1
  - 總帳：2023.1.1047-1
  - 網頁檢視器：2023.1.16  
388-1

### 錯誤修正

- 修復從登錄節點以活動目錄用戶身份提交作業失敗的問題。這個問題是由於與頭節點上的外部 Active Directory 整合的組態不完整所造成。
- 重構 CloudFormation 範本平行處理器IAM原則 .yaml 中定義的原則，以防止因原則超出限制而造成的部署失敗。  
ParallelCluster API IAM
- 修正當頭節點在寫入金鑰時所花費的時間超過預期時，登入節點無法啟動的問題。

有關變更的詳細資訊，請參閱 ( 詳見 ) 的 [aws-parallelcluster-ui](#)套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 推出使用者介面版本

2024年2月8日

變更：

- 更新了 Lambda 運行時環境到 Python V3.9

有關變更的詳細資訊，請參閱 ( 詳見 ) 的 [aws-parallelcluster-ui](#) 套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 已發佈使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。

2023 年 12 月 21 日

功能：

- 增加了對使用私有網絡PCUI部署的支持。
- 增加了選擇性地將權限邊界應用於由PCUI和PCAPI基礎架構創建的每個IAM角色的可能性
- 增加了選擇性地將前綴應用於由和PCAPI基礎結構創建的每個IAM角色PCUI和策略的可能性。
- 增加了對 3.8.0 ParallelCluster 版本的支持，在嚮導中沒有功能奇偶校驗。

有關變更的詳細資訊，請參閱 ( 詳見 ) 的 [aws-parallelcluster-ui](#)套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 版本 3.8.0 已發佈](#)

AWS ParallelCluster 版本  
3.8.0 已發布。

2023 年 12 月 19 日

增強功能：

- 為 ML 添加對 Amazon EC2 容量區塊的支援。
- 添加對洛基 Linux 8 的支援，通過過build-image 程CustomAmi 創建。目前沒有公共官方的 AWS ParallelCluster Rocky8 Linux AMI 可用。
- 新增Scheduling/ScalingStrategy 參數以控制針對Slurm運算節點啟動 Amazon EC2 執行個體時要使用的叢集擴展策略。可能的值為all-or-nothing、greedy-all-or-nothing、best-effort，並且all-or-nothing 為預設值。
- 新增HeadNode/SharedStorageType 參數以使用EFS儲存，而不是從主節點根磁碟區NFS匯出叢集內部共用檔案系統資源：ParallelCluster、Intel、Slurm 和資料。/home此增強功能可減少頭部節點網路的負載。
- 允許通過配置文件的FSx 部SharedStorage 分掛

載/home為EFS或外部共享存儲。

- 新增參數SlurmSettings/MungeKeySecretArn 以允許使用來自 AWS Secrets Manager 的外部使用者定義MUNGE金鑰。
- 新增Monitoring/Alarms/Enabled 參數以切換叢集的 Amazon CloudWatch 警示。
- 新增頭部節點警示以監控 Amazon 運作EC2狀態檢查、CPU使用率和頭節點的整體狀態，並將其新增至使用叢集建立的 CloudWatch 儀表板。
- 新增對資料儲存庫關聯的支援，PERSISTENT\_2 DeploymentType 為 Lustre 管FSx理使用。
- 新增Scheduling/SlurmSettings/Database/DatabaseName 參數以允許使用者指定資料庫伺服器上資料庫的自訂名稱，以用於Slurm計算。
- CapacityReservationTarget/CapacityReservationId 在計算資源中進行設定時，請InstanceType 建立選用的組態參數。

- 新增為建立的IAM角色和原則指定前置詞的可能性 AWS ParallelCluster API。
- 新增可能性，以指定要套用至由建立的IAM角色和原則的權限界限 AWS ParallelCluster API。

## 變更

- 升級Slurm至二月二十二日
- 將NVIDIA驅動程式升級至版本
- 將CUDA工具組升級至 12.2.2 版。
- 使用開放原始碼NVIDIAGPU驅動程式 (OpenRM) 作為 Linux 的NVIDIA核心模組，而非NVIDIA封閉原始碼模組。
- 取代新的Scheduling/ScalingStrategy 叢集all\_or\_nothing\_batch 配置，在Slurm簡歷程式中移除配置參數的支援。
- 將叢集警示命名慣例變更為「[叢集名稱]-[元件名稱]-[指標]」。
- 針對根EBS磁碟區和其他磁碟ADC區，將區域中的預設磁碟區類型從 gp2 變更為 gp3。
- 的選用權限界限現在 AWS ParallelCluster API會套用至

API基礎結構建立的每個IAM角色。

- 將EFA安裝程式升級至1.29.1.
- EFA-驅動程序：  
efa-2.6.0-1
- EFA-配置：efa-config-1.15-1
- 外匯基金會概況：efa-profile-1.5-1
- 利布-AWS：libfabric-aws-1.19.0-1
- RDMA 核心：rdma-core-46.0-1
- 打開MPI：openmpi40-aws-4.1.6-1
- 在所有支援的版本中升級GDRCopy至 2.4 版OSes，但使用 2.3.1 版的 CentOS 7 除外。
- 升級aws-cfn-bootstrap 至 2.0-28 版本。
- 在 aws-parallelcluster-batch-cli中添加對 Python 3.10 的支持。

#### 錯誤修正

- 修改 Compute Resources 中宣告的執行個體類型清單時，修正叢集更新復原後不一致的擴展配置問題。



- 修正透過叢集組態檔整合至外部LDAP伺服器的叢集中切換沒有 root 權限的使用者時產生使用者SSH金鑰的問題。
- 修復設置時禁用省Slurm電模式的問題Scaledown Idletime = -1 。
- 修正Slurm會計update\_slurm\_database\_password.sh 指令碼中安Slurm裝目錄的硬式編碼路徑。

[AWS ParallelCluster 版本 3.7.2 已發布](#)

AWS ParallelCluster 版本 3.7.2 已發布。

2023 年 10 月 25 日

變更：

- 升級Slurm至 2.

## [AWS ParallelCluster 已發佈使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。 2023 年 10 月 20 日

### 功能：

- 增加了對 ParallelCluster 3.7.2 的支持，嚮導中的功能奇偶校驗僅限於FSx文件緩存和基於內存的計劃與多個實例類型的兼容性。

### 錯誤修正：

- 已修正當PCUI沒有與 Cost Explorer 互動的權限時，造成 UI 錯誤的問題。

### 改善項目

- 通過將訪問令牌TTL從 10 分鐘減少到 5 分鐘來提高安全性。

有關變更的詳細資訊，請參閱 ( 詳見 ) 的[aws-parallelcluster-ui](#)套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 版本](#)

### [3.7.1 已發佈](#)

AWS ParallelCluster 版本  
3.7.1 已發布。

2023 年 9 月 22 日

變更：

- 升級Slurm至二點五 ( 從 23 點到 4 點 )
  - 將混音器升級到 4.2.6 ( 從 3.2.3 ) 。
  - 將資料庫升級至 1.15.3 ( 從 1.12.0 ) 。
- 將EFA安裝程式升級至1.26.1，修正 P5 中的 RDMA寫入資料問題。
  - EFA-驅動程序：
    - efa-2.5.0-1
  - EFA-配置：
    - efa-config-1.15-1
  - EFA-個人資料：
    - efa-profile-1.5-1
  - 利物件庫-AWS：
    - libfabric-aws-1.18.2-1
  - ERdma-核心：
    - rdma-core-46.0-1 .
  - 打開MPI：
    - openmpi40-aws-4.1.5-4 。

## [AWS ParallelCluster 版本](#) [3.7.0 已發佈](#)

AWS ParallelCluster 版本  
3.7.0 已發布。

2023 年 8 月 30 日

增強功能：

- 使用組態檔案 Support 在計算資源中 AWS ParallelCluster 設定靜態和動態節點優先順YAML序。
- 添加對 Ubuntu 22 的支持。RSA默認情況下不支持密鑰。
- 新增佇列組態設定，`JobExclusiveAllocation` 以便在任何指定時間將分割區中的節點專屬配置給單一工作。
- 允許在叢集建立和叢集更新時間覆寫`aws-parallelcluster-node` 套件。對於頭節點，這適用於叢集更新。僅用於開發目的。
- 避免NFS伺服器在運算節點上啟動。
- 添加對登錄節點的支持。
- 在為Slurm計算資源指定多個執行個體類型時，允許以記憶體為基礎的排程。
- 新增將現有 Amazon 檔案快取掛接為共用儲存的支援。

變更：

- 依預設，為Slurm動態節點指派 1000 的優先順序 (權重)。這樣，Slurm可以優先處理閒置靜態節點，而不是閒置動態節點。
- 讓aws-parallelcluster-node 守護進程僅處 AWS ParallelCluster 理託管 Slurm分區。
- 將EFS-utils 看門狗輪詢間隔增加到 10 秒。此變更會在設定EncryptionInTransit 為時套用true，這是導致監視器執行的唯一條件。
- 將EFA安裝程式升級至1.25.1。
  - EFA-驅動程序：  
efa-2.5.0-1 (來自) efa-2.1.1g
  - EFA-配置：efa-config-1.15-1 (來自) efa-config-1.13-1
  - EFA-概況:efa-profile-1.5-1 (沒有變化)
  - 利布-AWS：libfabric-aws-1.18.1-0 (來自) libfabric-aws-1.17.1-1
  - RDMA 核心：rdma-core-46.0-1 (來自) rdma-core-43.0-1

- 打開MPI : openmpi40  
-aws-4.1.5-4 ( 從  
openmpi40-aws-4.1.  
5-1 )
- 升級Slurm至版本
- 將 Imds/ 的默認值ImdsSupport 從 1.0 版更改為 2.0 版。
- 棄用 Ubuntu 18.
- 將預設的根磁碟區大小更新為 40 GB , 以考慮 CentOS 7 的限制。
- 限制頭節點內文件 /tmp/wait\_condition\_handle.txt 的權限 , 以便只有 root 才能讀取它。
- 建立一個Slurm分割區節點清單對應JSON檔案 , 供節點套件精靈使用 , 以辨識 PC 管理的分割區和符號。Slurm
- 將NVIDIA驅動程式升級至版本 535.54.03。
- 將CUDA程式庫升級至 12.2.0 版。
- 升級NVIDIA結構管理器到製造商經理 -535.
- 僅將 ARM PL 升級至版本 23.04.1。
- 升級NICEDCV至版本2023.0-15487 。
  - 伺服器 : 2023.0.15487-1
  - xdcv : 2023.0.551-1

- 總帳：2023.0.1039-1
- 網頁檢視器：2023.0.15487-1

#### 錯誤修正：

- 將驗證添加到Scaledown Idletime 值，以防止設置小於 -1 的值。
- 使用 Ubuntu 深度學習AMI在 DCV啟用的GPU執行個體上修正叢集建立失敗。
- 修正使 CustomLambdaRole 用建立 ParallelCluster CloudFormation 自訂資源提供者時造成懸置IAM政策的問題。
- 修正使用 SlurmSettings/Dns/UseEc2Hostnames equals 時，在具有多個網路介面的執行個體上造成運算節點DNS名稱錯位的問題 True

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。 [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[只有文件版本](#)

AWS ParallelCluster 版本 3 特定的用戶指南發布。 2023 年 7 月 17 日

文檔只有版本：

- AWS ParallelCluster 版本 3 有自己獨立的用戶指南。



## [AWS ParallelCluster 版本](#)

### [3.6.1 已發佈](#)

AWS ParallelCluster 版本  
3.6.1 已發布。

2023 年 7 月 5 日

變更：

- `clustermgtd` 如果將運算節點新增至多個Slurm分割區，請避免重複所看到的節點。

錯誤修正：

- 刪除根卷設備名稱的硬編碼 ( `/dev/sda1` 和 `/dev/xvda` ) 並從期間使用的AMI ( `s` ) 中檢索它 `create-cluster` 。
- 修正使用ElasticIp 設定為的 CloudFormation 自訂資源時叢集建立失敗True。
- 修正使用具有大型組態檔案的 AWS CloudFormation 自訂資源時，叢集建立和更新失敗的問題。
- 修正防止在 Ubuntu 上停用ptrace保護功能，且不允許在 libfabric 中使用跨記憶體附加 (CMA) 的問題。
- 修正使用多個執行個體類型且不傳回執行個體時，快速容量不足的容錯移轉邏輯。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#)和

套CHANGELOG 件的檔案。[aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。 2023 年 6 月 7 日

### 變更：

- 已將預設 AWS ParallelCluster API 版本升級至 3.6.0。

### 錯誤修正：

- 修正 AWS GovCloud (美國西部) 區域中斷的部署問題。
- 分割面板現在會在建立開始後正確載入叢集詳細資料。

### 備註：

- 中不提供「成本監控」功能 AWS GovCloud (US) Regions。

有關變更的詳細資訊，請參閱 ( 詳見 ) 的 [aws-parallelcluster-ui](#) 套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 版本 3.6.0 發布](#)

AWS ParallelCluster 版本  
3.6.0 發布。

2023 年 5 月 22 日

文件：

- 新增的文件[AWS ParallelClusterPython 函式庫](#)。

增強功能：

- 添加支持RHEL8.
- 新增用於建立和管理叢集的[AWS CloudFormation 自訂資源](#) CloudFormation。
- 在[Slurm配置文件中添加對自定義叢群](#) AWS ParallelCluster 配YAML置的支持。
- 建Slurm置支援LUA.
- 將每個叢集的佇列數目上限從 10 增加到 50。每個佇列最多可以有 50 個運算資源。每個叢集最多可以有 50 個運算資源。
- 添加支持為OnNodeStart、OnNodeConfigured 和OnNodeUpdated 參數中配置的事件指定多個[自定義操作腳本](#)的序列。
- 新增設定區段HealthChecks /Gpu，以便在工作執行之前在計算節點上套用GPU健康狀態檢查。

- 在SlurmQueues 和SlurmQueues /ComputeResources 組態Tags中新增對的支援。
- 在配置[DetailedMonitoring](#) 中添加Monitoring 對的支持。
- 在 AWS ParallelCluster [CloudWatch 儀表板](#) 中新增mem\_used\_percent disk\_used\_percent 標頭節點記憶體和根磁碟區磁碟使用率追蹤的指標，並設定警示以監控這些指標。
- 新增 AWS ParallelCluster 受管理記錄檔的記錄輪替支援。
- 在[CloudWatch 儀表板](#)中追蹤常見的運算節點錯誤和動態節點最長閒置時間。
- 強制DCV驗證器伺服器在建立通訊SSL端時至少使用TLS-1.2通訊協定。
- 在所有受支援的作業系統上安裝[NVIDIA資料中心GPU管理員 \(DCGM\)](#) 套件，除了aarch64centos7和alinux2。
- 預設載入核心模組 [nvidia-uvms](#)，為驅動程式提供統一虛擬記憶體 (UVM) 功能。CUDA
- 將[NVIDIA持續性常駐程式](#)安裝為系統服務。

**變更：**

- 升級Slurm至版本 23.02.2 (從版本22.05.8)。
- 升 munge 到版本0.5.15 (從版本0.5.14)。
- 將設定SlurmTreeWidth 為 30。
- 將 Slurmprolog和epilog配置/opt/slurm/etc/scripts/epilog.d/ 分別設置為目標目錄/opt/slurm/etc/scripts/prolog.d/ 。
- 在計算節點註冊期間，執行Prolog指令碼的最長設SlurmBatchStartTimeout 定為 3 分鐘。
- 將 CloudWatch 記錄檔RetentionInDays 的預設值從 14 天增加到 180 天。
- 將EFA安裝程式升級至1.22.1。
  - Dms : 2.8.3-2
  - EFA-驅動程序 : efa-2.1.1g (沒有變化)
  - EFA-配置 : efa-config-1.13-1 (沒有變化)
  - EFA-概況:efa-profile-1.5-1 (沒有變化)

- 利布-AWS : libfabric-aws-1.17.1-1 (來自) libfabric-aws-1.17.0-1
- RDMA 核心 : rdma-core-43.0-1 (沒有變更)
- 打開MPI : openmpi40-aws-4.1.5-1 (不變)
- 將光澤客戶端版本升級到 Amazon Linux 2 2.12 上。光澤客戶端 2.12 已經安裝在 Ubuntu 20.04、18.04 和 CentOS > = 7.7 上。
- 把光澤的客戶端版本升級至 CentOS 2.10.8 7.6 上。
- 將 NVIDIA 驅動程式升級至版本 470.182.03 (從版本 470.141.03 )。
- 將 NVIDIA 網狀架構管理員升級至版本 470.182.03 (從版本 470.141.03 )。
- 將 NVIDIA CUDA 工具包升級到版本 11.8.0 (從版本 11.7.1 )。
- 將 NVIDIA CUDA 範例升級至版本 11.8.0。
- 將英特爾 MPI 庫升級到 2021 年更新 9 版 (從 2021 年版本更新 6 )。如需詳細資訊，請參閱 [Intel® MPI 媒體櫃 2021 年更新 9](#)。
- 升級 NICEDCV 至版本 2023.0-15022 (從版本 2022.2-14521 )。

- 服務器：2023.0.15  
022-1（來自  
版本2022.2-14  
521-1）。
- xdcv：2023.0.54  
7-1（來自版  
本2022.2.519-1）。
- gl：2023.0.10  
27-1（來自版  
本2022.2.1012-1）。
- 瀏覽器：2023.0.15  
022-1（來自  
版本2022.2.14  
521-1）。
- 升級aws-cfn-b  
ootstrap 至版  
本2.0-24。
- 為 AWS Batch 叢集建置容器  
映像時，升級 CodeBuild 環  
境使用的映像檔：
  - aws/codebuild/  
amazonlinux2-  
x86\_64-st  
andard:4.0（從aws/  
codebuild/amaz  
onlinux2-x86\_64-  
standard:3.0）。
  - aws/codebuild/amaz  
onlinux2-aarch64-  
standard:2  
.0（從aws/  
codebuild/amaz  
onlinux2-aarch64-  
standard:1.0）。

**錯誤修正：**

- 修復 Amazon EFS 和 Amazon FSx 網路安全群組驗證程式，以避免報告錯誤。
  -
- 修正 build-image 作業期間缺少由 Image Builder 建立的資源標記。
- 修復更新策略，MaxCount 以始終對 MaxCount 屬性執行數值比較。
- 修正具有多張網路卡的運算節點執行個體上的 IP 對齊。
- 修正執行佇列參數更新且 Slurm 帳戶組態未更新 slurm\_parallelcluster\_slurmdbd.conf 時 StoragePass 中的取代。
- 修正使用現有 EFS 檔案系統建立叢集時，造成懸置安全性群組的問題。
- 修復重新啟動後導致 cfn-hup 守護進程失敗的問題。
- 將帶有 INVALID\_REG 標誌的動態節點視為 Slurm 受保護模式的引導程序失敗。Slurm 註冊失敗的靜態節點在之後已被視為啟動程序失敗 node\_replacement\_timeout 。



如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。[aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。

2023 年 5 月 16 日

### 增強功能：

- 從 AWS ParallelCluster 版本 3.6.0 開始，添加對 RHEL 8 的支持。
- 新增叢集成本監控。
- 從 AWS ParallelCluster 版本 3.6.0 開始，增加佇列和計算資源配額。

### 變更：

- 改進了叢集建立精靈的使用者介面。
- 提高了 AWS ParallelCluster UI 部署的速度。
- 改進了添加新用戶的介面。
- 依預設，佇列位於頭節點子網路中。

### 錯誤修正：

- 叢集建立完成後，切換至正確的區域。
- 修復「編輯群集」功能中的加載指示器顯示。
- 修正移除EBS SnapshotId 屬性時的叢集建立。

有關變更的詳細資訊，請參閱  
(詳見) 的 [aws-parallelcluster-ui](#) 套CHANGELOG 件檔案  
GitHub。

## [AWS ParallelCluster 推出使用者介面版本](#)

AWS ParallelCluster 使用者介面版本已發佈。

2023 年 4 月 17 日

### 增強功能：

- 叢集建立精靈重新設計。
- 重新設計叢集記錄頁面。
- 為共享存儲添加自定義名稱設置。
- 在將儲存裝置新增至叢集時，新增多個儲存選擇。
- 添加DeletionPolicy 對 Amazon EFS 和光澤FSx的支持。
- 在群集配ImdsSupport 置中添加設置。
- 新增對 C7 執行個體類型的支援。
- 添加了教程[還原至先前的 AWS系統管理員文件版本](#)。

### 變更：

- 叢集配YAML置大小最大為 1MB。
- 由於使用 Boto3 IAM 臨時憑據的授權，用戶未註銷。
- 在選取執行個HPC體時停用多執行緒選項。
- 刪除了在集群創建頁面上禁用回滾。

- 在提供所需資訊之前，使用者無法使用 AWS ParallelCluster UI。
- 最多可新增 10 個佇列。
- 在 AWS ParallelCluster UI 安裝期間不會覆寫SSM-SessionManager RunShell 文件。

錯誤修正：

- 修復損壞的重置密碼鏈接。
- 修復delete stack由於EcrPrivateRepository 不為空而導致的損壞
- 修復了多個用戶管理屬性部分中生成SSH密鑰複選框的初始化問題。
- 修復了具有未定義屬性的作業導致的崩潰。
- 固定SCRATCHFSx設置。
- 修復了啟動和停止實例按鈕，單擊一次後仍然啟用。

有關變更的詳細資訊，請參閱 ( 詳見 ) 的[aws-parallelcluster-ui](#)套CHANGELOG 件檔案 GitHub。

## [AWS ParallelCluster 版本](#)

### [3.5.1 已發布](#)

AWS ParallelCluster 版本  
3.5.1 發布。

2023 年 3 月 29 日

增強功能：

- 新增獨立pclusterCLI[安裝程式可執行檔](#)。

變更：

- 將EFA安裝程式升級至1.22.0.
  - EFA-驅動程序：  
efa-2.1.1g (來自) efa-2.1.1-1
  - EFA 配置：(來自 EFA 配置 -1.12 efa-config-1.13-1 -1)
  - EFA-概況:efa-profile-1.5-1 (沒有變化)
  - 利布-AWS：libfabric-aws-1.17.0-1 (來自) libfabric-aws-1.16.1amzn3.0-1
  - RDMA 核心：rdma-core-43.0-1 (沒有變更)
  - 打開MPI：openmpi40-aws-4.1.5-1 (從openmpi40-aws-4.1.4-3 )

升級NICEDCV至版本2022.2-14521 。

- 伺服器：2022.2.14521-1
- xdcv：2022.2.519-1
- 總帳：2022.2.1012-1
- 網頁檢視器：2022.2.14521-1

#### 錯誤修正：

- 修正叢集更新過程中，移除共用 Amazon EBS 磁碟區之間MountDir和移除/etc/exports 時的模式比對造成的可能節點啟動失敗。
- 修正以防止在每clustermgtd 次反覆運算時截斷compute\_console\_output 記錄檔案。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。[aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 版本](#) [3.5.0 發布](#)

AWS ParallelCluster 版本  
3.5.0 發布。

2023 年 2 月 20 日

增強功能：

- 使[AWS ParallelCluster 用 UI](#)存取和管理叢集。
- 在您可以在工作負載中參考的 AWS ParallelCluster CloudFormation 範本中新增已建立版本的原則。
- 添加一個 AWS ParallelCluster Python 庫，您可以使用自己的代碼。
- CloudWatch 在運算節點啟動程序失敗時，將運算節點主控台輸出的記錄新增至 Amazon。
- 將包含失敗代碼和原因的失敗字段添加到集群創建失敗時describe-cluster 輸出。
- 添加驗證器以防止在調用子進程模塊時惡意字符串注入。
- 如果叢集狀態在佈建靜態節點PROTECTED 時變更為，則叢集建立失敗。

變更：

- 升級至Slurm版本 22.05.8 (從版本22.05.7)



- 將EFA安裝程式升級至1.21.0.
  - EFA-驅動程序：  
efa-2.1.1-1 (來自) efa-2.1
  - EFA 配置：(來自 EFA 配置 -1.11-efa-config-1.12-1 1)
  - EFA-概況:efa-profile-1.5-1 (沒有變化)
  - 利布-AWS：libfabric-aws-1.16.1amzn3.0-1 (來自) libfabric-aws-1.16.1
  - RDMA 核心：rdma-core-43.0-1 (來自) rdma-core-43.0-2
  - 打開MPI：openmpi40-aws-4.1.4-3 (不變)
- 使Slurm控制器日誌更加詳細，並為省Slurm電插件啟用其他日誌記錄。

#### 錯誤修正：

- 啟用Slurm帳戶時，請確認叢集名稱不超過 40 個字元，以修正叢集資料庫的建立。
- 修正 Amazon EC2 執行個體狀態檢查失敗時導致重新啟動運算節點而被取代的問題。clustermgtd Slurm
- 修正因為頭節點上的IAM原則不正確而導致運算節點 (其

他帳戶共用容量保留) 無法啟動的問題。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#)、[aws-parallelcluster-cookboo](#)  
[kaws-parallelcluster-node](#)、  
和套CHANGELOG 件的檔案。[aws-parallelcluster-ui](#)  
GitHub

### [AWS ParallelCluster 版本](#) [3.4.1 發布](#)

AWS ParallelCluster 版本  
3.4.1 發布。

2023 年 1 月 13 日

錯誤修正：

- 修正Slurm排程器問題，此問題可能會導致計算節點的內部登錄中不正確的更新應用程式。因此，如果發生此問題，EC2執行個體可能無法使用，或可能由不正確的執行個體類型提供支援。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#)和  
套CHANGELOG 件的檔案。[aws-parallelcluster-](#)  
[cookbookaws-parallelcluster-](#)  
[node](#) GitHub

## [AWS ParallelCluster 版本](#) [3.4.0 已發佈](#)

## AWS ParallelCluster 版本 3.4.0 發布。

2022 年 12 月 22 日

### 增強功能：

- 新增跨多個可用區域啟動節點的支援，以提高容量可用性。
- 新增為每個佇列指定多個子網路的支援，以提高容量可用性。
- 在 [Iam/](#) 中添加新的配置參數，[ResourcePrefix](#) 以指定由創建的 IAM 資源的路徑和名稱的前綴 AWS ParallelCluster。
- 新增新的組態區段 [DeploymentSettings](#) [\\_LambdaFunctionsVpcConfig](#)，以指定 AWS ParallelCluster Lambda 函數使用的 Vpc 組態。
- 新增指定要在叢集更新期間在 head 節點中執行的自訂指令碼的功能。[OnNodeUpdated](#) 當使用 Slurm 作為調度程序時，可以使用 [HeadNodeCustomActions](#) //指定腳本。

### 變更：

- 移除針對現有檔案系統建立 Amazon EFS 掛載目標的作業。
- 使用掛載EFS檔案系統amazon-efs-utils。EFS可以使用傳輸中加密和IAM授權使用者來掛載檔案系統。
- 在美分OS7和 Ubuntu 上安裝軟件 5.67 以支持EFS傳輸中加密。
- 將EFA安裝程式升級到 1.20.0 (來源1.18.0)。
  - EFA-驅動程序：  
efa-2.1 (來自) efa-1.16.0-1
  - EFA-配置：efa-config-1.11-1 (沒有變化)
  - EFA-概況:efa-profile-1.5-1 (沒有變化)
  - 利布-AWS：libfabric-aws-1.16.1 (來自) libfabric-aws-1.16.0~amzn4.0-1
  - RDMA 核心：rdma-core-43.0-2 來自() rdma-core-41.0-2
  - 打開MPI：openmpi40-aws-4.1.4-3 從 (openmpi40-aws-4.1.4-2 )
- 升級Slurm至版本 22.05.7 (來源22.05.5)。

- 將 Python 升級到3.9.16和3.7.16. (來自3.9.15和3.7.13)。
- 使用時  
Slurm22.05.7 , IDLE  
+CLOUD+COMPLETING  
+POWER\_DOWN+NOT  
\_RESPONDING 狀態中的  
動態節點不會被視為健康狀  
態不良。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。[aws-parallelcluster-cookbookaws-parallelcluster-node](#) [GitHub](#)

## [AWS ParallelCluster 版本](#)

### [3.3.1 發布](#)

## AWS ParallelCluster 版本

### 3.3.1 發布。

2022 年 12 月 2 日

#### 變更：

- Amazon EC2 棄用兩年後，現在可以使用官方 AWS ParallelCluster 產品 AMIs。
- 將 AWS ParallelCluster API Lambda 的記憶體大小增加到 2048，以減少冷啟動處罰並避免逾時。

#### 錯誤修正：

- 防止替換 Lustre 檔案系統 FSx 的 Managed 檔案系統，以及包含對計算叢集子網路 ID 變更的叢集更新中的資料遺失。
- [SharedStorage](#) DeletionPolicy 適用於叢集更新動作。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集套CHANGELOG](#) 檔案。  
GitHub

[AWS ParallelCluster 文檔只有 hpc6id 注意事項](#)

AWS ParallelCluster 僅文件更新

2022 年 12 月 2 日

- AWS ParallelCluster 不支援/設定的 hpc6id 執行個體類型。[HeadNodeInstanceType](#)

## [AWS ParallelCluster 版本](#)

### [3.1.5 發布](#)

## AWS ParallelCluster 版本

### 3.1.5 發布。

2022 年 11 月 16 日

#### 增強功能：

- 修正Slurm防止閒置節點終止的問題。
- 將EFA安裝程式升級至 1.18.0
  - EFA-驅動程序：  
efa-1.16.0-1
  - EFA-配置：efa-config-1.11-1 (來自) efa-config-1.9-1
  - EFA-概況:efa-profile-1.5-1 (沒有變化)
  - 利布-AWS：libfabric-aws-1.16.0~amzn4.0-1 (來自)。libfabric-1.13.2
  - RDMA 核心：rdma-core-41.0-2 (來自) rdma-core-37.0
  - 打開MPI：openmpi40-aws-4.1.4-2 (從openmpi40-aws-4.1.1-2 )

#### 變更：

- 添lambda:ListTags 加lambda:Un



tagResource 到ParallelClusterUserRole 用於集群更新的 AWS ParallelCluster API堆棧。

- 將英特爾MPI庫升級到 2021 版更新 6 ( 從 2021 年版本更新 4 ) 。如需詳細資訊，請參閱 [Intel® MPI 媒體櫃 2021 年更新 6](#)。
- 將NVIDIA驅動程式升級至版本
- 將NVIDIA網狀架構管理員升級至版本 470.141.03 (從 1 年起)。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。[aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 版本](#) [3.3.0 發布](#)

## AWS ParallelCluster 版本 3.3.0 發布。

2022 年 11 月 2 日

### 增強功能：

- 當Slurm作為排程器使用時，新增對運算資源的多個執行個體配置設定的支援。如需詳細資訊，請參閱[使用Slurm 進行多個實例類型分配](#)。
- 使用更新的配置添加對[SharedStorage](#) 透過叢集更新進行新增和移除的支援。如需詳細資訊，請參閱[共用儲存](#)。
- DeletionPolicy 為[Efs](#)和[FsxLustre](#) 共用儲存設定新增組態參數，以支援儲存區保留。
- 使用新的配置參數 [Scheduling /SlurmSettings](#) /添加對Slurm會計的支持[Database](#)。如需詳細資訊，請參閱[Slurm會計與AWS ParallelCluster](#)。
- 新增對隨需容量保留 (ODCR) 和容量保留資源群組的支援。如需詳細資訊，請參閱[使用隨需容量保留 \(ODCR\) 啟動執行個體](#)。
- 新增組態參數以指定叢集中要支援的IMDS版本，或在叢集、/和

build、[Imds/ImdsSupport](#)、組態中建置映像基礎結構。[ImdsImdsSupport](#)

- [PlacementGroup](#) 在 [Networking /ComputeResources](#) 部分添加對 [SlurmQueues](#) /的支持。
- 新增對具有多個網路介面 (ENI每個裝置僅限一個網路介面) 的執行個體的支援。
- 透過檢查連接的安全群組中的CIDR區塊，改善外部 Amazon EFS 檔案系統的聯網驗證。
- 新增驗證器以檢查已配置的執行個體類型是否支援放置群組。
- 將執行NFS緒設定為最小值 (256，最大值 (8，核心 \* 4))，以確保更好的穩定性和效能。
- 在建置階段移動NFS安裝，以減少組態時間。
- 為部署時建立的 EcrImageBuilder SNS主題啟用伺服器端加密，AWS ParallelCluster API並用於在 docker 映像建置事件上通知。

變更：

- 更改 [SlurmQueues //Networking PlacementGroup](#) /的行為為Enabled。現在，它會為

每個計算資源建立唯一的受管置放群組，而不是針對所有運算資源建立單一受管置放群組。

- 添加對 [SlurmQueues //Networking Placement Group /Name](#) 作為首選命名方法的支持。
- 將頭節點標籤從 Launch Template 移至執行個體定義，以避免在標籤更新時取代標頭節點。
- 禁用通過啟動模板中的設置執行 `cloud-init` 和不通過 `CpuOptions` 設置執行的腳本進行多線程。
- 在 API 基礎架構、API 碼頭容器和叢集 Lambda 資源中，將 Python 升級至 3.9 版，並將 NodeJS JS 升級至第 16 版。
- 刪 Python 對 `aws-parallelcluster-batch-cli`。
- 升級 Slurm 至版本 22.05.5 (來源 21.08.8-2)。
- 將 NVIDIA 驅動程式升級至版本 470.141.03 (來源 470.129.06)。
- 將 NVIDIA 網狀架構管理員升級至版本 470.141.03 (來源 470.129.06)。

- 將NVIDIACUDA工具組升級至 11.7.1 (from 11.4.4) 版。
- 將 AWS ParallelCluster 虛擬環境中使用的 Python 從升級到。 3.7.13 3.9.15
- 將EFA安裝程式升級至 1.18.0 版。
  - EFA-驅動程序：  
efa-1.16.0-1 ( 沒有變化 )
  - EFA-配置： ( ) efa-config-1.11-1 from efa-config-1.10-1
  - EFA-概況:efa-profile-1.5-1 (沒有變化)
  - 利布-AWS : libfabric-aws-1.16.0~amzn4.0-1 ( 來自 )。 libfabric-aws-1.16.0~amzn2.0-1
  - RDMA 核心 : rdma-core-41.0-2 (來自) rdma-core-37.0
  - 打開MPI : openmpi40-aws-4.1.4-2 ( 從openmpi40-aws-4.1.1-2 )
- 升級NICEDCV至版本 2022.1-13300 (來源2022.0-12760 )。

- 啟用的 `SingleSubnetValidator` 制 `Queues`。
- 當 `DRAIN` 節點處於 `COMPLETING` 狀態時，請勿取代節點，因為 `Eilog` 可能仍在執行中。

錯誤修正：

- 修正在傳遞不正確的篩選器時，`AWS ParallelCluster ListClusterLogStreams` 命令中篩選器參數的驗證失敗。
- 當與其他 [SharedStorage / EfsSettings](#) 參數 `FileSystemId` 一起指定時，修正 [EfsSettings](#) 參數 `SharedStorage` / 驗證失敗。以前，`FileSystemId` 不包括在內。
- 修正變更組態中的 [SharedStorage](#) 其他變更順序時的叢集更新。
- 修正 `UpdateParallelClusterLambdaRole` 了 `AWS ParallelCluster API` 將日誌上傳到 `CloudWatch`。
- 修正 `Cinc` 在執行任何食譜之前安裝套件時不使用本機 `CA` 憑證套件的問題。
- 在設置時修復升級 `ubuntu pcluster build-image`

時Build:UpdateOsPack  
ages:Enabled:true 的  
問題。

- 通過重複密鑰失敗來修復  
YAML群集配置的解析。

如需變更的詳細資訊，請  
參閱上的 [aws-並行叢集](#)和  
套CHANGELOG 件的檔  
案。[aws-parallelcluster-  
cookbookaws-parallelcluster-  
node](#) GitHub

[AWS ParallelCluster 文檔只  
API參考添加。](#)

AWS ParallelCluster 僅文件更  
新

2022 年 10 月 27 日

- 添加了版本 3 [AWS ParallelC  
luster API 參考資料](#) 的文  
檔。

## [AWS ParallelCluster 版本](#)

### [3.2.1 已發布](#)

## AWS ParallelCluster 版本

### 3.2.1 發布。

2022 年 10 月 3 日

#### 增強功能：

- 改善將主機路由表與不同網路卡建立關聯的邏輯，以更好地支援具有多個 Amazon EC2 執行個體的執行個體 NICs。

#### 變更：

- 將NVIDIA驅動程式升級至版本
- 將NVIDIA網狀架構管理員升級至版本 470.141.03。
- 停用cron工作作業 man-dbmllocate，這可能會對節點效能產生負面影響。
- 將英特爾MPI庫升級到 2021.6.0.602。
- 因應此安全性風險，請將 Python 從 3.7.10 升級至 3.7.13。

#### 錯誤修正：

- 避免叢集配置無法使用DescribeCluster 時失敗。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#)和



套CHANGELOG 件的檔  
案。[aws-parallelcluster-  
cookbookaws-parallelcluster-  
node](#) GitHub

## [AWS ParallelCluster 版本 3.2.0 發布](#)

AWS ParallelCluster 版本  
3.2.0 發布。

2022 年 7 月 27 日

增強功能：

- 在 Slurm 中添加對[基於內存的調度](#)的支持。
  - 在 Slurm 叢集配置中設定運算節點實際記憶體。
  - 添加新的配置參數 [Scheduling / SlurmSettings / EnableMemoryBasedScheduling](#) 以在 Slurm 中啟用基於內存的調度。
  - 新增配置參數 [Scheduling / SlurmQueues / ComputeResources / SchedulableMemory](#) 以覆寫排程器在計算節點上看到的記憶體的預設值。
- 提高叢集配置更新的靈活性，以避免在可能的情況下停止和啟動整個叢集。添加新的配置參數 [Scheduling / SlurmSettings / QueueUpdateStrategy](#) 以設置當計算節點需要更新和更換配置時使用的首選策略。
- 在 Amazon EC2 執行個體發生容量不足問題時，改善

可用運算資源的容錯移轉機制。當節點啟動因為容量不足而失敗時，依可設定的時間停用運算節點。

- 添加支持掛載現有[FSx for ONTAP](#)和[FSx for OpenZFS](#)文件系統。
- 新增支援，以掛載現有[Amazon 彈性檔案系統的多個執行個體](#)，FSx適用於[Lustre ONTAP](#)、FSx用於[開放ZFS](#)檔案系統。
- 在建立新檔案系統時，新增FSx對[Lustre 永久性\\_2 部署類型](#)的支援。
- 使用[pcluster configure](#) 精靈時，提示使EFA用者啟用支援的執行個體類型。
- 新增使用 Slurm 重新啟動運算節點的支援。
- 改善Slurm電源狀態的處理，同時考量節點的手動關閉電源。
- 在產品中安裝 NVIDIA GDRCopy 2.3 AMIs 以啟用低延遲GPU記憶體複製。

變更：

- 將EFA安裝程式升級至 1.17.2 版。
  - EFA驅動程序：efa-1.16.0-1

- EFA配置 : efa-confi  
g-1.10-1
- EFA設定檔 : efa-profi  
le-1.5-1
- 布料:libfabric-  
aws-1.16.0~amzn2.  
0-1
- RDMA核心 : rdma-core  
-41.0-2
- 打開MPI : openmpi40-  
aws-4.1.4-2
- 升級NICEDCV至版本
- 將NVIDIA驅動程式升級至版本
- 將NVIDIA網狀架構管理員升級至版本 470.129.06。
- 在根EBS磁碟區和其他磁碟區中，將預設磁碟區類型從gp2 變更為 gp3。
- 對於由以下FSx人員建立的Lustre 檔案系統的變更：  
AWS ParallelCluster
  - 將預設部署類型變更  
為Scratch\_2 。
  - 將 Lustre 伺服器版本變更為。2.12
- 傳遞現有[Enabled](#)的[PlacementGroup](#) /true時  
不需要設置Placement  
Group /Id。
- Id當Placement  
Group /Enabled被  
明確設置為時，不允

許設置Placement

Group /false。

- 將parallelcluster:cluster-name 標籤新增至由建立的所有資源 AWS ParallelCluster。
- 添加lambda:ListTags 和lambda:UntagResource ParallelClusterUserRole 用於集群更新的 AWS ParallelCluster API堆棧。
- 僅在啟用配置參數HeadNode/Imds/時，限制IMDS對根和叢集管理員使用者Secured的IPv6存取權。
- 使用自訂AMI，請使用AMI根磁碟區大小而非 ParallelCluster預設值 35 GiB。可以在叢集配置檔案中變更該值。
- 當組態參數Scheduling //SlurmQueues ComputeResources /SpotPrice 低於所需的最低 Spot 請求履行價格時，會自動停用運算叢集。
- 在更新期間加入或移除區段時，顯示requested\_value 和變更集中的current\_value 值。

- 停用深度學習中可用的aws-ubuntu-eni-helper 服務AMIs，以避免在設定具有多張網路卡的執行個體configure\_nw\_interface.sh 時發生衝突。
- 刪除對 Python 3.6 的支持。
- 使MTU用多張網路卡設定執行個體時，所有網路介面的設定為 9001。
- 設定運算節點時移除尾端點 FQDN。
- 管理中的靜態節點POWERING\_DOWN 。
- 不取代中的動態節點，POWER\_DOWN 因為作業可能仍在執行中。
- 只有在叢集配置中更新Scheduling 參數時，才會在叢集更新時重新啟動clustermgtd 和slurmctld 精靈。
- 更新slurmctld 和slurmdsystemd服務文件。
- 僅在啟用配置參數HeadNode/Imds/時，限制IMDS對根和叢集管理員使用者Secured的IPv6存取權。
- 設定Slurm組態AuthInfo=cred\_expire=70 以減少重新呼叫的工作必須等待的時間，才能在節點無法使用時重新啟動。

- 升級第三方食譜依賴關係：
  - 四月份 -7.4.2 (由四月至七月四日起)
  - 第 4.5.2 行 ( 從第 4.0.1 行開始 )
  - 開放式 -2.10.3 ( 來自開放時間 -2.9.1 )
  - 火焰病毒 -3.5.1 ( 來自比恩夫 -3.4.2 )
  - 性腺素 -6.0.4 ( 來自塞利諾克斯 -3.1.1 )
  - 尤姆 -7.4.0 (從尤姆 -6.1.1)
  - 尤姆-埃珀爾 -4.5.0 ( 來自尤姆-埃帕爾 -4.1.2 )

#### 錯誤修正：

- 修復默認行為以在構建自定義時跳過 AWS ParallelCluster 驗證和測試步驟AMI。
- 修復文件句柄洩漏computemgtd 。
- 修正偶爾導致已啟動的執行個體因為回應中尚未提供而立即終止的爭用情形。EC2 DescribeInstances
- 修正使用 Arm 處理器對執行個體類型DisableSimultaneousMultithreading 參數的支援。
- 修復從以前版本升級時 AWS ParallelCluster API堆棧更新失敗。在中新增用於ListImagePipelineI

images 動作的資源模式EcrImageDeletionLambdaRole 。

- 修正建立FSx適用於 Lustre 檔案系統時，AWS ParallelCluster API新增從 Amazon S3 匯入或匯出所需遺失的權限。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。[aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub



## [AWS ParallelCluster 今年迄今為止僅限文件的更新](#)

AWS ParallelCluster 僅限文件更新。

2022 年 7 月 6 日

新章節：

- [最佳做法：預算提醒V3](#)
- [最佳做法：將叢集移至新的次要版本或 AWS ParallelCluster 修補程式版本V3](#)
- [使用 Amazon S3V3](#)
- [使用 競價型執行個體V3](#)
- [Slurm叢集保護模式V3](#)
- [AWS ParallelCluster 資源和標記V3](#)
- [Amazon CloudWatch 儀表V3](#)
- [與 Amazon Amazon Amazon CloudWatch 的整合V3](#)
- [Elastic Fabric AdapterV3](#)
- [AWS ParallelCluster AMI 定制V3](#)
- [使用隨需容量保留 \(ODCR\) 啟動執行個體V3](#)
- [AMI 修補和 Amazon EC2 實例替換V3](#)
- [AWS ParallelCluster 的運作方式V3](#)
- [使用 AWS KMS 金鑰設定共用儲存區加密V3](#)
- [在多佇列模式叢集中執行工作V3](#)

- [應用 AWS ParallelCluster 程式介面V3](#)

區段更新：

- [最佳做法：網路效能V3](#)：添加了使用彈性織物適配器的最佳實踐。
- [AWS Identity and Access Management 權限 AWS ParallelClusterV3](#)：各種更新和添加[使用亞馬遜 FSx 進行光澤時的其他用 AWS ParallelCluster pcluster 戶政策](#)。
- [AWS ParallelCluster 疑難排V3](#)：各種更新。

## [AWS ParallelCluster 版本](#)

### [3.1.4 發布](#)

## AWS ParallelCluster 版本

### 3.1.4 發布。

2022 年 5 月 16 日

#### 增強功能：

- 如果密碼不存在，則添加 [Directory Service / PasswordSecretArn](#) 失敗的驗證。

添加對啟用JWT身份驗證Slurm的支持。

#### 變更：

- 升級Slurm至第二版。
- 建置Slurm與JWT支援。
- 傳遞現有[Enabled的 PlacementGroup /true](#)時不需要設置Placement Group /Id。
- 添加lambda:TagResource到ParallelClusterUserRole 用於集群創建和映像創建的ParallelCluster API堆棧。

#### 錯誤修正：

- 修正使用export-cluster-logs 指令搭配--filters 選項時，匯出叢集記錄的功能。

- 修復 AWS Batch Docker 入口點以使用/home共享目錄來協調多節點並行作業執行。
- 將Slurm健康狀態不良的靜態節點設定為關閉時，請重設節點位址，以避免將容量不足的靜態節點視為啟動程序失敗節點。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。 [aws-parallelcluster-cookbookaws-parallelcluster-node](#) [GitHub](#)

## [AWS ParallelCluster 版本](#)

### [3.1.3 發布](#)

## AWS ParallelCluster 版本

### 3.1.3 發布。

2022 年 4 月 20 日

#### 增強功能：

- 當切換到另一個用戶以及以另一個用戶身份執行命令時，可以在創建HOME目錄時執行SSH密鑰創建，例如在SSH登錄期間，同時執行密鑰創建。
- 在組態參數 [Directory Service](#) /中新增對FQDN與LDAP辨別名稱的支援[DomainName](#)。新的驗證程式現在會檢查這兩種語法。
- 在頭節點上部署的新 `update_directory_service_password.sh` 指令碼支援手動更新SSSD組態中的 Active Directory 密碼。密碼由 AWS Secrets Manager as 從叢集配置中擷取。
- 新增在沒有預設值的環境中部署API基礎結構的支援VPC。

#### 變更：

- 在 x86\_64 官方中禁用更深層次的 C 狀態AMIs並通過 `build-image` 命令AMIs

創建，以確保高性能和低延遲。

- 作業系統套件更新和安全性修正。
- 將 Amazon Linux 2 基本映像檔變更為AMIs搭配核心 5.10 使用。

錯誤修正：

- 由於新的 Image Builder 策略，在圖像構建成功DELETE\_FAILED 後修復構建EC2映像堆棧。
- 修復包含多個域地址時的配置參數 [Directory Service /DomainAddr](#) 轉換為 ldap\_uri SSSD 屬性。

如需變更的詳細資訊，請參閱 [aws-並行叢集](#) 的CHANGELOG 檔案以及上的套件。 [aws-parallelcluster-cookbook](#) GitHub

## [AWS ParallelCluster 版本](#)

### [3.1.2 發布](#)

## AWS ParallelCluster 版本

### 3.1.2 發布。

2022 年 3 月 2 日

#### 變更：

- 升級Slurm至版本 21.08.6 (來源21.08.5)。

#### 錯誤修正：

- 修正當叢集部署在沒有網際網路存取的子網路中時，計算節點上的/etc/hosts 檔案更新。
- 修正運算節點啟動程序，以便在加入叢集之前等待暫時磁碟機初始化。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 套CHANGELOG 件的檔案。

GitHub

## [AWS ParallelCluster 版本](#)

### [3.1.1 版本發布](#)

## AWS ParallelCluster 版本

### 3.1.1 發布。

2022 年 2 月 10 日

- 透過與管理的使用 [中目錄 \(AD\) 網域整合](#)，新增對多個使用者叢集環境的支援 AWS Directory Service。
- 在叢集配置檔案 [UseEc2Hostnames](#) 中新增對的支援。當設定為 true 時，運算節點使用 Amazon EC2 預設主機名稱 (例如 ip-1-2-3-4)。
- 添加對在 [沒有 Internet 訪問權限的子網絡](#) 中創建集群的支持。
- 新增每個佇列對多個運算執行個體類型的支援。
- 新增對使用資 NVIDIA 訊卡 ARM 執行個體 Slurm 進行 GPU 排程的支援。
- 將 `(-n)`、`(-c)` 和 `(-i)` 的縮寫旗標加入 AWS ParallelCluster CLI 至 `aws parallelcluster create --region region --image-configuration image-configuration --new-changed-deleted new-changed-deleted` 的縮寫旗標加入 AWS ParallelCluster CLI 至 `aws parallelcluster create --region region --image-configuration image-configuration --new-changed-deleted new-changed-deleted`。
- 添加對 Lustre [AutoImportPolicy](#) 參數 `NEW_CHANGED_DELETED` 選項 FSx 的支持。
- 將 `parallelcluster:compute-resource-`



name 標籤新增至運算節點使用的EC2LaunchTemplates 資源。

- 改善叢集內建立的安全群組，以便在為某些頭節點和/或佇列指定SecurityGroups 參數時，允許來自自訂安全性群組的輸入連線。
- 安裝的NVIDIA驅動程式和程式CUDA庫ARM。

變更：

- 升級Slurm至版本 21.08.5 (來源20.11.8)。
- 將Slurm外掛程式升級至版本 21.08 (自20.11)。
- 升級NICEDCV至版本 2021.3-11591 (來源2021.1-10851 )。
- 將NVIDIA驅動程式升級至版本 470.103.01 (來源470.57.02 )。
- 將NVIDIA網狀架構管理員升級至版本 470.103.01 (來源470.57.02 )。
- 升級CUDA至版本 11.4.4 (來源11.4.0)。
- [英特爾MPI](#)已更新至 2021 年版本更新 4 (自 2019 年版本更新 8 更新)。如需詳細資訊，請參閱 [Intel® MPI 媒體櫃 2021 年更新 4](#)。
- 升級PMIx至版本 3.2.3 (來源3.1.5)。

- 移除將失敗的運算節點傾印至 `/home/logs/compute`。運算節點日誌檔可在 Amazon EC2 主控台日誌中 CloudWatch 和中取得。
- 啟用潛在的抑制 `SlurmQueues` 和 `ComputeResources` 長度驗證器。
- 在 Amazon Linux 2 上的執行個體啟動時停用套件更新。
- 建立 AWS ParallelCluster 自訂映像時停用 Amazon EC2 ImageBuilder 增強型影像中繼資料。
- 明確地將 `cloud-init` 數據源設置為。EC2 這樣可以節省 Ubuntu 及 CentOS 平台的開機時間。
- 在計算叢集啟動範本名稱中使用計算資源名稱而非執行個體類型。
- 將 `stderr` 和標準輸出重定向到 CLI 日誌文件，以防止 `pcluster` 輸出中不需要的文本。CLI
- 將配置/安裝食譜移動到從主食譜中調用的食譜分開。保持現有的入口點並向後兼容。
- 在 AMI 建置期間下載 Intel HPC 平台的相依性，以避免叢集建立期間與網際網路連線。
- 設定 Slurm 節點時，請勿 - 從計算資源名稱中刪除。

- 未安裝NVIDIA驅動程序GPU式Slurm時，請勿在中進行配置。
  - 修復中的ecs:ListContainerInstances 權限BatchUserRole 。
  - 修正未指定前置詞 (先前匯出為前置None詞) 時叢集記錄的匯出問題。
  - 修正叢集更新失敗時未執行的復原問題。
  - 修復中的ecs:ListContainerInstances 權限BatchUserRole 。
  - 如果指定了不支援，則HeadNode透過引發錯誤來修正KmsKeyId正的RootVolume 結構描述。
  - 修復要在 CloudWatch 儀表中顯示的 Amazon FSx 丟失的指標。
  - 修復EfaSecurityGroupValidator 。
- 先前，在提供並EFA啟用自訂安全性群組時，可能會產生錯誤的失敗。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集](#) 和套CHANGELOG 件的檔案。 [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 版本](#)

### [3.0.3 發布](#)

## AWS ParallelCluster 版本

### 3.0.3 發布。

2022年1月17日

- 停用 Amazon Linux 2 上的log4j-cve-2021-44228-hotpatch 代理程式 (Log4jHotPatch )，以避免可能的效能降低。有關更多信息，請參閱 [Amazon Linux 修補程序公告阿帕奇 Log 4j](#)。

如需變更的詳細資訊，請參閱上的 [aws-並行叢集CHANGELOG](#) 檔案和套件。[aws-parallelcluster-cookbook](#) GitHub

## [AWS ParallelCluster 版本](#)

### [3.0.2 發布](#)

AWS ParallelCluster 版本  
3.0.2 發布。

2021 年 11 月 5 日

將[Elastic Fabric Adapter](#)安裝  
程式升級至 1.14.1

- EFA配置 : efa-confi  
g-1.9-1 ( 從efa-confi  
g-1.9 )
- EFA配置文件 : efa-profi  
le-1.5-1 ( 從efa-  
profile-1.5 )
- EFA內核模塊 : efa-1.14.  
2 ( 來自efa-1.13.0 )
- RDMA核心 : rdma-core  
-37.0 ( 從rdma-core  
-35 )
- 利布 : libfabric  
-1.13.2 ( 來  
自libfabric-1.13.0 )
- 打開MPI : openmpi40-  
aws-4.1.1-2 ( 不變 )

GPUDirectRDMA如果執行個  
體類型支援，則永遠會啟用。  
組[GdrSupport](#)態選項沒有任何  
作用。

如需變更的詳細資訊，請參閱  
[aws-並行叢集](#)的CHANGELOG  
檔案以及上的套件。[aws-  
parallelcluster-cookbookaws-  
parallelcluster-node](#) GitHub

## [AWS ParallelCluster 版本](#)

### [3.0.1 發布](#)

AWS ParallelCluster 版本  
3.0.1 發布。

2021 年 10 月 27 日

#### 叢集配置移轉工具

- 客戶現在可以將其叢集配置從第 2 AWS ParallelCluster 版格式移轉YAML為基礎的 AWS ParallelCluster 版本 3 格式。如需詳細資訊，請參閱[pcluster3-config-converter](#)。

#### 頭節點可以停止

- 停止運算叢集之後，可以停止頭節點，然後再使用 Amazon EC2 主控台或[停止執行個體](#) AWS CLI 命令重新啟動。

#### 預設從~/.aws/config 檔案 AWS 區域 讀取

- 對於指[pcluster](#)令，如果 AWS 區域 未在組態檔案、環境或指令行中指定，則會使用~/.aws/config 檔案[default] 區段中region設定中 AWS 區域 指定的預設值。

如需變更的詳細資訊，請參閱[aws-並行叢集](#)的CHANGELOG 檔案以及上的套件。[aws-](#)

[parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 版本](#) [3.0.0 發布](#)

AWS ParallelCluster 版本  
3.0.0 發布。

2021 年 9 月 10 日

### Support 透過 Amazon API 閘道進行叢集管理

- 客戶現在可以使用 Amazon API 閘道透過HTTP端點管理和部署叢集。這為腳本或事件驅動的工作流程開闢了新的可能性。

指 AWS ParallelCluster 命令行介面 (CLI) 也經過重新設計，以便與此相容性，API並包含新的JSON輸出選項。這項新功能可讓客戶使用以及實作類似的建置區塊功CLI能。

### 改善自訂AMI建立

- 客戶現在可以存取更強大的程序，以便AMIs使用 EC2 Image Builder 建立和管理自訂。Custom 現在AMIs可以透過單獨的 AWS ParallelCluster 規劃檔進行管理，並且可以使用指[pcluster build-image](#) 命令行介面中的 AWS ParallelCluster 指令建立。

如需變更的詳細資訊，請參閱 [aws-並行叢集](#)的CHANGELOG



檔案以及上的套件。 [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

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。