



AWS ParallelCluster 使用者指南 (v3)

AWS ParallelCluster



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

Copyright © 2026 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

什麼是 AWS ParallelCluster	1
AWS ParallelCluster 運作方式	2
AWS ParallelCluster 程序	2
clustermgtd	2
clusterstatusmgtd	3
computemgtd	3
AWS 使用的 服務 AWS ParallelCluster	3
Amazon API Gateway	4
AWS Batch	5
CloudFormation	5
Amazon CloudWatch	5
Amazon CloudWatch Events	5
Amazon CloudWatch Logs	6
AWS CodeBuild	6
Amazon DynamoDB	6
Amazon Elastic Block Store	6
Amazon Elastic Compute Cloud	7
Amazon Elastic Container Registry	7
Amazon EFS	7
Amazon FSx for Lustre	7
Amazon FSx for NetApp ONTAP	8
Amazon FSx for OpenZFS	8
AWS Identity and Access Management	8
AWS Lambda	8
Amazon RDS	9
Amazon Route 53	9
Amazon Simple Notification Service	9
Amazon Simple Storage Service	9
Amazon VPC	10
Elastic Fabric Adapter	10
EC2 Image Builder	10
Amazon DCV	10
AWS ParallelCluster 內部目錄	10
設定 AWS ParallelCluster	12

先決條件	12
設定 AWS 帳戶	12
建立金鑰對	14
安裝 AWS ParallelCluster CLI	14
AWS ParallelCluster 在虛擬環境中安裝 (建議)	14
使用 pip AWS ParallelCluster 在非虛擬環境中安裝	16
將安裝 AWS ParallelCluster 為獨立應用程式	17
安裝後要採取的步驟	19
安裝 AWS ParallelCluster UI	19
安裝 PCUI	20
堆疊參數	22
設定自訂網域	26
Amazon Cognito 使用者集區選項	27
識別 AWS ParallelCluster 和 PCUI 版本	29
PCUI 成本	30
開始使用	30
使用 CLI AWS ParallelCluster 設定和建立叢集	31
使用 UI AWS ParallelCluster 設定和建立叢集	40
連接至叢集	41
多個使用者存取叢集	42
建立 Active Directory	43
建立具有 AD 網域的叢集	43
登入與 AD 網域整合的叢集	46
執行 MPI 任務	47
AWS Managed Microsoft AD 透過 LDAP(S) 叢集組態的範例	47
最佳實務	51
最佳實務：前端節點執行個體類型選取	51
最佳實務：網路效能	51
最佳實務：預算提醒	53
最佳實務：將叢集移至新的 AWS ParallelCluster 次要或修補程式版本	53
從 AWS ParallelCluster 2.x 移至 3.x	54
自訂引導操作	54
AWS ParallelCluster 2.x 和 3.x 使用不同的組態檔案語法	54
包容性語言	60
排程器支援	61
AWS ParallelCluster CLI	61

IMDS 組態更新	64
使用 AWS ParallelCluster	65
AWS ParallelCluster UI	66
AWS Lambda 中的 VPC 組態 AWS ParallelCluster	67
AWS Identity and Access Management 中的 許可 AWS ParallelCluster	68
AWS ParallelCluster Amazon EC2 執行個體角色	69
AWS ParallelCluster 範例 pcluster 使用者政策	70
AWS ParallelCluster 用於管理 IAM 資源的使用者範例政策	84
AWS ParallelCluster 用於管理 IAM 許可的組態參數	91
網路組態	105
AWS ParallelCluster 在單一公有子網路中	106
AWS ParallelCluster 使用兩個子網路	108
AWS ParallelCluster 在單一私有子網路中，使用 連線 AWS Direct Connect	109
AWS ParallelCluster 使用 AWS Batch 排程器	110
AWS ParallelCluster 在沒有網際網路存取的單一子網路中	112
佈建的登入節點 AWS ParallelCluster	118
登入節點的安全性	118
登入節點的網路	119
登入節點的儲存體	119
登入節點的 Imds 屬性	119
登入節點生命週期	120
執行登入節點集區所需的許可	120
自訂引導操作	121
Configuration	123
引數	127
具有自訂引導操作的範例叢集	127
如何更新 IMDSv2 自訂引導指令碼的範例	129
如何更新 IMDSv1 組態的範例	129
使用 Amazon S3	130
範例	130
使用 競價型執行個體	131
案例 1：沒有執行中任務的 Spot 執行個體遭到中斷	132
案例 2：執行單一節點任務的 Spot 執行個體遭到中斷	132
案例 3：執行多節點任務的 Spot 執行個體遭到中斷	132
支援的排程器 AWS ParallelCluster	132
Slurm Workload Manager	132

AWS Batch	189
共用儲存	197
Amazon EBS	199
Amazon EFS	200
FSx for Lustre	201
FSx for ONTAP、FSx for OpenZFS 和檔案快取	201
使用共用儲存	202
配額	205
標記	206
檢視標籤	206
監控 AWS ParallelCluster 和日誌	209
與 Amazon CloudWatch Logs 的整合	210
Amazon CloudWatch 儀表板	213
叢集指標的 Amazon CloudWatch 警示	215
AWS ParallelCluster 設定的日誌輪換	217
pcluster CLI 日誌	218
Amazon EC2 主控台輸出日誌	218
擷取 PCUI 和 AWS ParallelCluster 執行時間日誌	219
擷取和保留日誌	221
AWS CloudFormation 自訂資源	224
託管的提供者堆疊 AWS ParallelCluster	225
叢集資源	226
叢集操作	229
對包含 AWS ParallelCluster 自訂資源的堆疊進行故障診斷	229
Elastic Fabric Adapter	230
預設 EFA 網路組態	231
自訂 EFA 網路介面	231
啟用 Intel MPI	231
AWS ParallelCluster API	233
AWS ParallelCluster API 文件	233
使用 部署 AWS ParallelCluster API AWS CLI	234
更新 API	236
叫用 AWS ParallelCluster API	237
存取 API 日誌和指標	239
AWS ParallelCluster 適用於 Terraform	239
透過 Amazon DCV 連接至前端和登入節點	240

Amazon DCV HTTPS 憑證	241
授權 Amazon DCV	241
使用 pcluster update-cluster	241
更新政策：定義	242
pcluster update-cluster 範例	245
AWS ParallelCluster AMI 自訂	248
AWS ParallelCluster AMI 自訂考量事項	248
執行自訂元件驗證測試	249
使用 pcluster 命令監控映像建置器程序，以協助偵錯	249
其他考量	250
使用隨需容量預留 (ODCR) 啟動執行個體	250
搭配 使用 ODCR AWS ParallelCluster	251
使用容量區塊 (CB) 啟動執行個體	259
搭配 使用 CB AWS ParallelCluster	259
AMI 修補和 Amazon EC2 執行個體替換	260
前端節點執行個體更新或取代	261
從暫時性磁碟機儲存資料	261
停止和啟動叢集的前端節點	262
作業系統	264
作業系統考量事項	264
的參考 AWS ParallelCluster	266
AWS ParallelCluster 第 3 版 CLI 命令	266
pcluster	267
pcluster3-config-converter	311
組態檔案	312
叢集組態檔案	313
建置映像組態檔案	438
AWS ParallelCluster API 參考	447
buildImage	448
createCluster	453
deleteCluster	459
deleteClusterInstances	462
deleteImage	464
describeCluster	467
describeClusterInstances	475
describeComputeFleet	479

describeImage	480
getClusterLogEvents	487
getClusterStackEvents	491
getImageLogEvents	495
getImageStackEvents	499
listClusters	503
listClusterLogStreams	507
listImageLogStreams	511
listImages	515
listOfficialImages	518
updateCluster	521
updateComputeFleet	527
AWS ParallelCluster Python 程式庫 API	529
AWS ParallelCluster Python 程式庫授權	530
安裝 AWS ParallelCluster Python 程式庫	530
叢集 API 操作	530
運算機群 API 操作	534
叢集和堆疊日誌操作	536
映像 API 操作	539
映像和堆疊日誌操作	541
範例	544
AWS Lambda 適用於 AWS ParallelCluster Python 程式庫	545
教學	547
在上執行您的第一個任務 AWS ParallelCluster	547
確認安裝	548
建立您的第一個叢集	548
登入您的頭部節點	549
使用 Slurm 執行您的第一個任務	550
建置自訂 AWS ParallelCluster AMI	551
如何自訂 AWS ParallelCluster AMI	552
建置自訂 AWS ParallelCluster AMI	552
修改 AWS ParallelCluster AMI	558
整合 Active Directory	560
建立 AD 基礎設施	562
(選用) 管理 AD 使用者和群組	577
建立叢集	579

以使用者身分連線至叢集	585
清除	586
使用 AWS KMS 金鑰設定共用儲存加密	591
建立政策	592
設定和建立叢集	593
在多個佇列模式叢集中執行任務	594
設定您的叢集	595
建立 叢集	596
登入前端節點	597
在多個佇列模式中執行任務	598
使用 AWS ParallelCluster API	601
使用Slurm會計建立叢集	615
步驟 1：建立的 VPC 和子網路 AWS ParallelCluster	616
步驟 2：建立資料庫堆疊	616
步驟 3：建立已啟用Slurm會計的叢集	616
使用外部Slurmdbd會計建立叢集	617
步驟 1：建立 Slurmdbd 堆疊	618
步驟 2：建立Slurmdbd已啟用的叢集	619
還原至先前的 AWS Systems Manager 文件版本	620
還原至先前的 SSM 文件版本	620
使用 建立叢集 CloudFormation	622
使用 CloudFormation 快速建立堆疊建立叢集	622
使用 CloudFormation 命令列界面 (CLI) 建立叢集	625
檢視 CloudFormation 叢集輸出	627
存取您的叢集	627
清除	628
使用 Terraform 部署 ParallelCluster API	628
定義 Terraform 專案	628
部署 API	631
所需的許可	631
使用 Terraform 建立叢集	634
定義 Terraform 專案	635
部署叢集	641
所需的許可	642
使用 Terraform 建立自訂 AMI	643
定義 Terraform 專案	643

部署 AMI	646
所需的許可	646
AWS ParallelCluster UI 與 Identity Center 整合	647
啟用 IAM Identity Center	648
將應用程式新增至 IAM Identity Center	649
使用 Pyxis 執行容器化任務	652
建立叢集	653
提交任務	655
使用啟用 EFA 的 FSx Lustre 建立叢集	655
需求	655
建立安全群組	656
建立檔案系統	657
建立叢集	658
使用 EFA 驗證 FSx 正在運作	659
使用 p6e-gb200 執行個體支援 NVIDIA-Imex	660
建立 Prolog 指令碼以管理 NVIDIA-Imex	660
建立 HeadNode OnNodeStart 自訂動作指令碼	664
建立叢集	665
驗證 IMEX 設定	666
使用啟動範本覆寫自訂運算節點網路介面	668
步驟 1：建立安全群組	668
步驟 2：建立啟動範本	669
步驟 3：建立具有啟動範本覆寫的叢集	671
AWS ParallelCluster 故障診斷	672
嘗試建立叢集	673
failureCode 是 OnNodeConfiguredExecutionFailure	673
failureCode 是 OnNodeConfiguredDownloadFailure	673
failureCode 是 OnNodeConfiguredFailure	674
failureCode 是 OnNodeStartExecutionFailure	674
failureCode 是 OnNodeStartDownloadFailure	674
failureCode 是 OnNodeStartFailure	675
failureCode 是 EbsMountFailure	675
failureCode 是 EfsMountFailure	675
failureCode 是 FsxMountFailure	676
failureCode 是 RaidMountFailure	676
failureCode 是 AmiVersionMismatch	676

failureCode 是 InvalidAmi	676
failureCode HeadNodeBootstrapFailure具有failureReason無法設定前端節點的。	677
failureCode 是 failureReason HeadNodeBootstrapFailure , 叢集建立逾時。	677
failureCode HeadNodeBootstrapFailure具有failureReason無法引導前端節點的。	678
failureCode 是 ResourceCreationFailure	679
failureCode 是 ClusterCreationFailure	679
在 CloudFormation 堆疊WaitCondition timed out...中查看	679
在 CloudFormation 堆疊Resource creation cancelled中查看	679
在 CloudFormation 堆疊中查看 Failed to run cfn-init...或其他錯誤	679
查看以 chef-client.log結尾 INFO: Waiting for static fleet capacity provisioning	680
查看 Failed to run preinstall or postinstall in cfn-init.log	680
在 CloudFormation 堆疊This AMI was created with xxx, but is trying to be used with xxx...中查看	680
在 CloudFormation 堆疊This AMI was not baked by AWS ParallelCluster...中查看	680
查看pcluster create-cluster命令無法在本機執行	680
其他支援	680
嘗試執行任務	680
srun 互動式任務失敗並發生錯誤 srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf	680
任務使用 squeue命令卡在 CF 狀態	681
執行大規模任務並查看 nfsd: too many open connections, consider increasing the number of threads in /var/log/messages	681
執行 MPI 任務	682
嘗試更新叢集	682
pcluster update-cluster 命令無法在本機執行	682
使用 clusterStatus UPDATE_FAILEDpcluster describe-cluster命令查看	683
叢集更新逾時	684
嘗試存取儲存體	684
使用外部 Amazon FSx for Lustre 檔案系統	684
使用外部 Amazon Elastic File System 檔案系統	684
嘗試刪除叢集	684
pcluster delete-cluster 命令無法在本機執行	684

叢集堆疊無法刪除	684
嘗試升級 AWS ParallelCluster API 堆疊	684
查看運算節點初始化中的錯誤	685
在 Node bootstrap error中查看 clustermgtd.log	685
我設定隨需容量保留 (ODCRs) 或區域預留執行個體	685
當我無法執行任務An error occurred (VcpuLimitExceeded)slurm_resume.log時 查看 ，或當我無法建立叢集時clustermgtd.log查看 。	687
當我無法執行任務An error occurred (InsufficientInstanceCapacity)slurm_resume.log時查看 ，或當我無法建立叢集 時clustermgtd.log查看 。	687
使用 查看節點處於 DOWN 狀態 Reason (Code:InsufficientInstanceCapacity)... ..	687
在 cannot change locale (en_US.utf-8) because it has an invalid name中查看 slurm_resume.log	687
上述案例都不適用於我的情況	688
叢集運作狀態指標疑難排解	688
請參閱執行個體佈建錯誤圖表	688
查看運作狀態不佳的執行個體錯誤圖表	690
請參閱運算機群閒置時間圖表	692
對叢集部署問題進行故障診斷	692
在 上檢視 CloudFormation 事件 CREATE_FAILED	693
使用 CLI 來檢視日誌串流	695
使用 重新建立失敗的叢集 rollback-on-failure	697
使用 Terraform 對叢集部署進行故障診斷	698
找不到 ParallelCluster API	698
使用者無權呼叫 ParallelCluster API	698
對擴展問題進行故障診斷	699
除錯的金鑰日誌	700
在我無法執行任務slurm_resume.log時看到InsufficientInstanceCapacity錯誤， 或在我無法建立叢集clustermgtd.log時看到錯誤	701
故障診斷節點初始化問題	702
對非預期的節點替換和終止進行故障診斷	704
取代、終止或關閉有問題的執行個體和節點	706
佇列 (分割區) Inactive 狀態	706
故障診斷其他已知節點和任務問題	706
置放群組和執行個體啟動問題	707

取代目錄	707
對 Amazon DCV 中的問題進行故障診斷	707
Amazon DCV 的日誌	707
Ubuntu Amazon DCV 問題	708
使用 AWS Batch 整合對叢集中的問題進行故障診斷	708
前端節點問題	709
運算問題	709
任務失敗	709
端點 URL 錯誤的連線逾時	709
對與 Active Directory 的多使用者整合進行故障診斷	709
Active Directory 特定疑難排解	710
啟用除錯模式	711
如何從 LDAPS 移至 LDAP	711
如何停用 LDAPS 伺服器憑證驗證	711
如何使用 SSH 金鑰而非密碼登入	712
如何重設使用者密碼和過期的密碼	712
如何驗證加入的網域	712
如何疑難排解憑證的問題	713
如何驗證與 Active Directory 的整合是否正常運作	715
如何對登入運算節點進行故障診斷	715
多使用者環境中 SimCenter StarCCM+ 任務的已知問題	716
使用者名稱解析的已知問題	716
如何解決主目錄建立問題	716
對自訂 AMI 問題進行故障診斷	717
在 cfn-hup 未執行時對叢集更新逾時進行故障診斷	718
網路故障診斷	719
單一公有子網路問題的叢集	719
onNodeUpdated 自訂動作上的叢集更新失敗	719
使用自訂Slurm組態查看錯誤	719
叢集警示	720
解決導致錯誤或失敗的作業系統組態變更	720
常見的作業系統組態問題	720
作業系統組態變更的最佳實務	724
AWS ParallelCluster 支援政策	725
安全	727
所使用服務的安全性資訊 AWS ParallelCluster	727

資料保護	728
資料加密	729
另請參閱	730
身分和存取權管理	730
法規遵循驗證	731
強制執行 TLS 1.2	731
判定目前支援的通訊協定	731
編譯 OpenSSL 和 Python	733
設定受限環境的安全群組	734
安全群組概觀	734
叢集操作所需的連接埠	734
建立自訂安全群組	736
在叢集組態中設定安全群組	737
在受限環境中使用 VPC 端點	738
安全群組組態的最佳實務	739
對安全群組問題進行故障診斷	739
支援的 AWS 區域	740
版本備註和文件歷史記錄	742
.....	dcclvi

什麼是 AWS ParallelCluster

AWS ParallelCluster 是一種 AWS 支援的開放原始碼叢集管理工具，可協助您在 中部署和管理高效能運算 (HPC) 叢集 AWS 雲端。它會自動設定所需的運算資源、排程器和共用檔案系統。您可以 AWS ParallelCluster 搭配 AWS Batch 和 Slurm 排程器使用。

透過 AWS ParallelCluster，您可以快速建置和部署概念驗證和生產 HPC 運算環境。您也可以在上建置和部署高階工作流程 AWS ParallelCluster，例如可自動化整個 DNA 排序工作流程的基因體入口網站。

您可以使用 AWS ParallelCluster 下列方法存取：

- [AWS ParallelCluster 命令列界面 \(CLI\)](#)
- [AWS ParallelCluster API](#)
- [PCUI](#) (隨 3.5.0 版新增)
- [AWS ParallelCluster Python 程式庫 API](#) (已新增 3.5.0 版)
- 作為 [AWS CloudFormation 自訂資源](#) (在 3.6.0 版中新增)

定價

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster 運作方式

AWS ParallelCluster 不僅是為了管理叢集而建置，也是為了做為如何使用 AWS 服務來建置 HPC 環境的參考。下列主題說明 AWS ParallelCluster 程序、AWS 服務 AWS ParallelCluster 使用的和方式，以及內部目錄。

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，會與基礎任務排程器 AWS ParallelCluster 互動，以管理運算節點佈建和移除。

對於以為基礎的 HPC 叢集 AWS Batch，AWS ParallelCluster 依賴提供的功能 AWS Batch 來管理運算節點。

clustermgtd

叢集管理常駐程式 (clustermgtd) 會執行下列任務：

- 清除非作用中分割區
- 管理與容量區塊相關聯的Slurm保留和節點（請參閱下一節）
- 管理靜態容量，以確保其始終正常運作
- 將排程器與 Amazon EC2 同步。
- 清除孤立的執行個體
- 在暫停工作流程之外發生的 Amazon EC2 終止時還原排程器節點狀態
- 管理運作狀態不佳的 Amazon EC2 執行個體（未通過 Amazon EC2 運作狀態檢查的執行個體）
- 管理排定的維護事件
- 管理運作狀態不佳的排程器節點（未通過排程器運作狀態檢查的節點）

與容量區塊相關聯的Slurm保留和節點管理

ParallelCluster 支援Machine Learning (CB) 的隨需容量保留 (ODCR) 和容量區塊。與 ODCR 不同，CB 可以有未來的開始時間，並且有時間限制。

`clustermgtd` 會搜尋迴圈中運作狀態不佳的節點、終止任何停機的 Amazon EC2 執行個體，如果它們是靜態節點，則會將其取代為新的執行個體。

AWS ParallelCluster 會以不同方式管理與容量區塊相關聯的靜態節點 – 即使 CB 尚未處於作用中狀態，也會建立叢集，並在 CB 處於作用中狀態時自動啟動執行個體。

與尚未啟用CBs 相關聯的運算資源對應的Slurm節點會保持在維護狀態，直到達到 CB 開始時間為止。這些Slurm節點會維持在與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任務，但任務會保持待定狀態，直到移除Slurm保留為止。

`clustermgtd` 會自動建立或刪除Slurm保留 – 它會根據 CB 狀態，將相關的 CB 節點置於維護狀態。當 CB 變為作用中時，會移除Slurm保留，節點會開始並可用於待定任務或提交新的任務。

達到 CB 結束時間時，節點會移回保留/維護狀態。當 CB 不再作用中且執行個體終止時，最多可以讓使用者將任務重新提交/重新排入佇列至新的佇列/運算資源。

`clusterstatusmgtd`

叢集狀態管理常駐程式 (`clusterstatusmgtd`) 會管理運算機群狀態更新。每分鐘都會擷取存放在 DynamoDB 資料表中的機群狀態，並管理任何 STOP/START 請求。

`computemgtd`

運算管理常駐程式 (`computemgtd`) 程序會在每個叢集運算節點上執行。每五 (5) 分鐘，運算管理協助程式會確認可以到達前端節點且運作狀態良好。如果經過五 (5) 分鐘，頭部節點無法到達或狀況不良，則會關閉運算節點。

AWS 使用的 服務 AWS ParallelCluster

使用下列 Amazon Web Services (AWS) 服務 AWS ParallelCluster。

主題

- [Amazon API Gateway](#)
- [AWS Batch](#)

- [CloudFormation](#)
- [Amazon CloudWatch](#)
- [Amazon CloudWatch Events](#)
- [Amazon CloudWatch Logs](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx for Lustre](#)
- [Amazon FSx for NetApp ONTAP](#)
- [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](#)
- [Amazon DCV](#)

Amazon API Gateway

Amazon API Gateway 是一項 AWS 服務，可讓您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket APIs

AWS ParallelCluster 使用 API Gateway 託管 AWS ParallelCluster API。

如需 Amazon API Gateway 的詳細資訊，請參閱 <https://aws.amazon.com/api-gateway/> : // 和 <https://docs.aws.amazon.com/apigateway/> : //。

AWS Batch

AWS Batch 是 AWS 受管任務排程器服務。它會動態佈建 AWS Batch 叢集中運算資源的最佳數量和類型（例如 CPU 或記憶體最佳化執行個體）。這些資源是根據批次任務的特定需求佈建，包括磁碟區需求。使用 AWS Batch，您不需要安裝或管理額外的批次運算軟體或伺服器叢集，即可有效執行任務。

AWS Batch 僅用於 AWS Batch 叢集。

如需詳細資訊 AWS Batch，請參閱 <https://aws.amazon.com/batch/> : // 和 <https://docs.aws.amazon.com/batch/> : //。

CloudFormation

CloudFormation 是一種 infrastructure-as-code 服務，提供在雲端環境中建模和佈建 AWS 和第三方應用程式資源的通用語言。這是使用的主要服務 AWS ParallelCluster。中的每個叢集 AWS ParallelCluster 都會以堆疊表示，而每個叢集所需的所有資源都會在 AWS ParallelCluster CloudFormation 範本中定義。在大多數情況下，AWS ParallelCluster CLI 命令會直接對應至 CloudFormation 堆疊命令，例如建立、更新和刪除。在叢集內啟動的執行個體會對啟動 AWS 區域 叢集的 CloudFormation 端點進行 HTTPS 呼叫。

如需詳細資訊 CloudFormation，請參閱 <https://aws.amazon.com/cloudformation/> : // 和 <https://docs.aws.amazon.com/cloudformation/> : //。

Amazon CloudWatch

Amazon CloudWatch (CloudWatch) 是一種監控和可觀測性服務，可為您提供資料和可行的洞見。這些洞見可用於監控您的應用程式、回應效能變更和服務例外狀況，以及最佳化資源使用率。在中 AWS ParallelCluster，CloudWatch 用於儀表板，以監控和記錄 Docker 映像建置步驟和 AWS Batch 任務的輸出。

在 2.10.0 AWS ParallelCluster 版之前，CloudWatch 僅用於 AWS Batch 叢集。

如需 CloudWatch 的詳細資訊，請參閱 <https://aws.amazon.com/cloudwatch/> 和 <https://docs.aws.amazon.com/cloudwatch/>。

Amazon CloudWatch Events

Amazon CloudWatch Events (CloudWatch Events) 提供近乎即時的系統事件串流，描述 Amazon Web Services (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 Events 用於 AWS Batch 任務。

如需 CloudWatch Events 的詳細資訊，請參閱 <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cwe-now-eb>。

Amazon CloudWatch Logs

Amazon CloudWatch Logs (CloudWatch Logs) 是 Amazon CloudWatch 的核心功能之一。您可以使用它來監控、存放、檢視和搜尋使用的許多元件的日誌檔案 AWS ParallelCluster。

在 2.6.0 AWS ParallelCluster 版之前，CloudWatch Logs 僅與 AWS Batch 叢集搭配使用。

如需詳細資訊，請參閱 [與 Amazon CloudWatch Logs 的整合](#)。

AWS CodeBuild

AWS CodeBuild (CodeBuild) 是一種 AWS 受管持續整合服務，可編譯原始程式碼、執行測試，並產生準備好部署的軟體套件。在中 AWS ParallelCluster，CodeBuild 用於在建立叢集時自動且透明地建置 Docker 映像。

CodeBuild 僅用於 AWS Batch 叢集。

如需 CodeBuild 的詳細資訊，請參閱 <https://aws.amazon.com/codebuild/> 和 <https://docs.aws.amazon.com/codebuild/>。

Amazon DynamoDB

Amazon DynamoDB (DynamoDB) 是一種快速且靈活的 NoSQL 資料庫服務。它用於存放叢集的最小狀態資訊。前端節點會追蹤 DynamoDB 資料表中的佈建執行個體。

DynamoDB 不會與 AWS Batch 叢集搭配使用。

如需 DynamoDB 的詳細資訊，請參閱 <https://aws.amazon.com/dynamodb/> 和 <https://docs.aws.amazon.com/dynamodb/>。

Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) 是一種高效能區塊儲存服務，可為共用磁碟區提供持久性儲存。所有 Amazon EBS 設定都可以透過組態傳遞。Amazon EBS 磁碟區可以初始化為空白，也可以從現有的 Amazon EBS 快照初始化。

如需 Amazon EBS 的詳細資訊，請參閱 <https://aws.amazon.com/ebs/> 和 <https://docs.aws.amazon.com/ebs/>。

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) 提供的運算容量 AWS ParallelCluster。前端和運算節點是 Amazon EC2 執行個體。可選取支援硬體虛擬機器 (HVM) 的任何執行個體類型。前端和運算節點可以是不同的執行個體類型。此外，如果使用多個佇列，也可以啟動部分或全部運算節點做為 Spot 執行個體。在執行個體上找到的執行個體存放區磁碟區會掛載為條紋邏輯磁碟區管理員 (LVM) 磁碟區。

如需 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 Docker 容器登錄檔，可讓您輕鬆儲存、管理和部署 Docker 容器映像。在中 AWS ParallelCluster，Amazon ECR 會存放建立叢集時所建置的 Docker 映像。然後，會使用 Docker 映像 AWS Batch 來執行所提交任務的容器。

Amazon ECR 僅用於 AWS Batch 叢集。

如需詳細資訊，請參閱 <https://aws.amazon.com/ecr/> 和 <https://docs.aws.amazon.com/ecr/>。

Amazon EFS

Amazon Elastic File System (Amazon EFS) 提供簡單、可擴展且全受管的彈性 NFS 檔案系統，可與服務和內部部署資源搭配使用 AWS 雲端。指定 [EfsSettings](#) 時，會使用 Amazon EFS。Amazon EFS 的支援已新增至 2.1.0 AWS ParallelCluster 版。

如需 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。[FsxLustreSettings](#) 屬性指定時，會使用 FSx for Lustre。2.2.1 AWS ParallelCluster 版中已新增 FSx for Lustre 的支援。

如需 FSx for Lustre 的詳細資訊，請參閱 <https://aws.amazon.com/fsx/lustre/> : // 和 <https://docs.aws.amazon.com/fsx/> : //。

Amazon FSx for NetApp ONTAP

FSx for ONTAP 提供建置在 NetApp 熱門 ONTAP 檔案系統上的全受管共用儲存系統。[FsxOntapSettings](#) 屬性 指定時會使用 FSx for ONTAP。3.2.0 AWS ParallelCluster 版中已新增 FSx for ONTAP 的支援。

如需 FSx for 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

FSx for OpenZFS 提供建置在熱門 OpenZFS 檔案系統上的全受管共用儲存系統。[FsxOpenZfsSettings](#) 屬性 指定時，會使用 FSx for OpenZFS。3.2.0 AWS ParallelCluster 版中已新增對 FSx for OpenZFS 的支援。

如需 FSx for OpenZFS 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 在中使用 (IAM) AWS ParallelCluster 為 Amazon EC2 為每個個別叢集特定的執行個體提供最低權限的 IAM 角色。AWS ParallelCluster 執行個體只能存取部署和管理叢集所需的特定 API 呼叫。

使用 AWS Batch 叢集時，也會為建立叢集時涉及 Docker 映像建置程序的元件建立 IAM 角色。這些元件包括允許在 Amazon ECR 儲存庫中新增和刪除 Docker 映像的 Lambda 函數。它們也包含允許刪除為叢集和 CodeBuild 專案建立之 Amazon S3 儲存貯體的函數。資源 AWS Batch、執行個體和任務也有角色。

如需 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) 會執行協調建立 Docker 映像的函數。Lambda 也會管理自訂叢集資源的清除，例如存放在 Amazon ECR 儲存庫和 Amazon S3 上的 Docker 映像。

如需 Lambda 的詳細資訊，請參閱 <https://aws.amazon.com/lambda/> 和 <https://docs.aws.amazon.com/lambda/>。

Amazon 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 使用適用於 AWS Batch 和 的 Amazon RDSSlurm。

如需 Amazon RDS 的詳細資訊，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 Route 53 的詳細資訊，請參閱 <https://aws.amazon.com/route53/> 和 <https://docs.aws.amazon.com/route53/>。

Amazon Simple Notification Service

(Amazon SNS) 是一種受管服務，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。

如需 Amazon SNS 的詳細資訊，請參閱 <https://aws.amazon.com/sns/> : // 和 <https://docs.aws.amazon.com/sns/> : //。

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) 會存放位於每個 中的 AWS ParallelCluster 範本 AWS 區域。AWS ParallelCluster 可設定為允許 CLI/SDK 工具使用 Amazon S3。

AWS ParallelCluster 也會在 中建立 Amazon S3 儲存貯體 AWS 帳戶，以存放叢集所使用的資源，例如叢集組態檔案。會在您建立叢集的每個 AWS 區域中 AWS ParallelCluster 維護一個 Amazon S3 儲存貯體。

當您使用 AWS Batch 叢集時，帳戶中的 Amazon S3 儲存貯體會用來儲存相關資料。例如，儲存貯體會存放從提交的任務建立 Docker 映像和指令碼時所建立的成品。

如需詳細資訊，請參閱 <https://aws.amazon.com/s3/> 和 <https://docs.aws.amazon.com/s3/>。

Amazon VPC

Amazon Virtual Private Cloud (VPC) 定義叢集中節點所使用的網路。

如需 Amazon VPC 的詳細資訊，請參閱 <https://aws.amazon.com/vpc/> : // 和 <https://docs.aws.amazon.com/vpc/> : //。

Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) 是執行個體的網路介面，可用來執行需要大規模進行高階節點間通訊的應用程式 AWS。

如需 Elastic Fabric Adapter 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，可協助您自動建立、管理和部署自訂、安全且 up-to-date 伺服器映像。

AWS ParallelCluster 使用映像建置器來建立和管理 AWS ParallelCluster 映像。

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/> : //。

Amazon DCV

Amazon DCV 是一種高效能遠端顯示通訊協定，提供在不同的網路條件下，將遠端桌面和應用程式串流交付至任何裝置的安全方式。指定 / [HeadNode 區段 Dcv](#) 設定時，會使用 Amazon DCV。Amazon DCV 的支援已新增至 2.5.0 AWS ParallelCluster 版。

如需 Amazon DCV 的詳細資訊，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 使用 在叢集內共用資料。下列目錄會在前端節點、運算節點和登入節點之間共用：

- /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

根據預設，這些目錄會在前端節點 EBS 磁碟區上建立，並在 NFS 匯出至運算和登入節點時共用。從 AWS ParallelCluster 3.8 開始，您可以透過將 [SharedStorageType](#) 參數設定為 `ef`，AWS ParallelCluster 讓 建立和管理 Amazon EFS 檔案系統來託管和共用這些目錄。當叢集向外擴展時，透過 EBS 磁碟區的 NFS 匯出可能會帶來效能瓶頸。使用 EFS，您可以避免在叢集橫向擴展時匯出 NFS，並避免與其相關聯的效能瓶頸。

設定 AWS ParallelCluster

下列主題說明如何設定 AWS ParallelCluster。您將了解如何安裝必要的工具，以及如何使用這些工具、如何實作和管理多個使用者對叢集的存取，以及最佳實務。

主題

- [先決條件](#)
- [安裝 AWS ParallelCluster 命令列界面 \(CLI\)](#)
- [安裝後要採取的步驟](#)
- [安裝 PCUI](#)
- [入門 AWS ParallelCluster](#)
- [多個使用者存取叢集](#)
- [最佳實務](#)
- [從 AWS ParallelCluster 2.x 移至 3.x](#)

先決條件

在開始設定和使用 之前 AWS ParallelCluster，請確定您已完成下列先決條件。

設定 AWS 帳戶

設定要使用 AWS 的帳戶 AWS ParallelCluster。

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

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

部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

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

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

建立具有管理存取權的使用者

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

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者 [AWS 管理主控台](#) 身分登入。在下一頁中，輸入您的密碼。

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

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

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

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

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

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

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

以具有管理存取權的使用者身分登入

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

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

指派存取權給其他使用者

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

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的[建立許可集](#)。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

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

建立金鑰對

若要部署叢集，AWS ParallelCluster 會啟動 Amazon EC2 執行個體來建立叢集前端節點和運算節點。若要執行叢集任務，例如執行和監控任務，或管理使用者，您必須能夠存取叢集前端節點。若要驗證您可以使用 SSH 存取前端節點執行個體，您必須使用 Amazon EC2 金鑰對。若要了解如何建立金鑰對，請參閱《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》中的[建立金鑰對](#)。

安裝 AWS ParallelCluster 命令列界面 (CLI)

AWS ParallelCluster 以 Python 套件形式分發，並使用 Python pip 套件管理員安裝。如需如何安裝 Python 套件的說明，請參閱 Python Packaging 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 pip AWS ParallelCluster 在非虛擬環境中安裝](#)
- [將安裝 AWS ParallelCluster 為獨立應用程式](#)

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

同時管理 AWS ParallelCluster 2 和 AWS ParallelCluster 3

如果您同時使用 AWS ParallelCluster 2 和 AWS ParallelCluster 3，並想要管理兩個套件 CLIs，建議您在不同[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 AWS ParallelCluster 和任何相關聯的叢集資源。

AWS ParallelCluster 在虛擬環境中安裝 (建議)

我們建議您 AWS ParallelCluster 在虛擬環境中安裝，以避免需求版本與其他 pip 套件衝突。

先決條件

- AWS ParallelCluster 需要 Python 3.7 或更新版本。如果您尚未安裝，請在 <https://python.org> 下載平台的[相容版本](#)。

在 AWS ParallelCluster 虛擬環境中安裝

1. 如果 virtualenv 未安裝，virtualenv 請使用安裝 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. 在虛擬環境中安裝 AWS ParallelCluster。

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

5. 安裝 Node Version Manager 和最新的長期支援 (LTS) Node.js version. AWS Cloud Development Kit (AWS CDK) requires Node.js for CloudFormation 以產生範本。

Note

如果您的 Node.js 安裝無法在平台上運作，您可以在最新的 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 AWS ParallelCluster 已正確安裝。

```
$ pcluster version
{
  "version": "3.15.0"
}
```

您可以使用 `deactivate` 命令來離開虛擬環境。每次啟動工作階段時，您必須[重新啟用環境](#)。

若要升級至最新版本的 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

您也可以使用 pip AWS ParallelCluster 在非虛擬環境中安裝，pip 是 Python 套件的套件管理員。

先決條件

- AWS ParallelCluster 需要 Python 3.7 或更新版本。如果您尚未安裝，請在 <https://python.org> 下載平台的[相容版本](#)。

安裝 AWS ParallelCluster

1. 使用 pip 進行安裝 AWS ParallelCluster。

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

當您使用 `--user` 切換時，會pip安裝 AWS ParallelCluster 到 `~/.local/bin`。

2. 安裝 Node Version Manager 和最新的長期支援 (LTS) Node.js version. AWS Cloud Development Kit (AWS CDK) requires Node.js for CloudFormation 以產生範本。

Note

如果您的 Node.js 安裝無法在平台上運作，您可以在最新的 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=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
```

3. 確認 AWS ParallelCluster 安裝正確。

```
$ pcluster version
{
  "version": "3.15.0"
}
```

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
```

將 安裝 AWS ParallelCluster 為獨立應用程式

在您的環境中將 安裝 AWS ParallelCluster 為獨立應用程式。請依照下節中的指示，在可用的作業系統 AWS ParallelCluster 上安裝。

先決條件

- 具有與可用 安裝程式版本相容的作業系統的環境。

Note

AWS ParallelCluster 需要 NodeJS. AWS ParallelCluster Installer 包含 NodeJS (v18) 的套件版本，如果尚未存在，則會安裝此套件。如果您的系統與 NodeJS v18 不相容，您應該在安裝之前安裝 NodeJS AWS ParallelCluster。

Linux

Linux x86 (64-bit)

在您的環境 AWS ParallelCluster 上安裝。

1. 下載最新的 [pcluster 安裝程式](#) ([檢查總和](#))。
2. 解壓縮安裝程式套件，並使用下列命令 AWS ParallelCluster 安裝：

```
$ unzip pcluster-installer-bundle-<VERSION>-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 AWS ParallelCluster 已正確安裝。

```
$ pcluster version  
{  
  "version": "3.15.0"  
}
```

對pcluster安裝錯誤進行故障診斷

- 如果未在步驟 4 中傳回 AWS ParallelCluster 版本，請重新啟動終端機或 source bash_profile來更新PATH變數，以包含新的二進位目錄，如下列範例所示：

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

- 如果您使用 `pcluster` 安裝來建立 `CustomActions` 指定為 HTTPS 資源而非 S3 URIs 叢集，您可能會看到 WARNING 訊息，指出這些資源可能無法驗證 (【SSL: CERTIFICATE_VERIFY_FAILED】)。這是由已知問題造成，如果您信任指定資源的真實性，您可以忽略此警告。

先前的安裝程式套件版本

- 無

安裝後要採取的步驟

本節假設您已安裝 AWS ParallelCluster。您將了解如何驗證 是否正確 AWS ParallelCluster 安裝、如何更新至最新版本的 AWS ParallelCluster，以及如何解除安裝。

您可以執行 來驗證 AWS ParallelCluster 是否正確安裝 [pcluster version](#)。

```
$ pcluster version
{
  "version": "3.15.0"
}
```

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

```
$ 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 PCUI

UI AWS ParallelCluster (PCUI) 是以 Web 為基礎的使用者介面，可鏡射 AWS ParallelCluster `pclusterCLI`，同時提供類似主控台的體驗。您可以在 中安裝和存取 PCUI AWS 帳戶。當您執行它時，PCUI 會存取 中 Amazon AWS ParallelCluster API Gateway 上託管的 API 執行個體 AWS 帳戶。Amazon API Gateway 如需 PCUI 的詳細資訊，請參閱 [AWS ParallelCluster UI](#)。

事前準備：

- 您必須擁有 AWS 帳戶
- 您必須能夠存取 AWS 管理主控台

如需詳細資訊，請參閱[設定 AWS 帳戶](#)。

主題

- [安裝 PCUI](#)
- [堆疊參數](#)
- [設定自訂網域](#)
- [Amazon Cognito 使用者集區選項](#)
- [識別 AWS ParallelCluster 和 PCUI 版本](#)
- [PCUI 成本](#)

安裝 PCUI

若要安裝 UI AWS ParallelCluster (PCUI) 的執行個體，請為您建立叢集 AWS 區域的 選擇一個 CloudFormation 快速建立連結。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 《使用者指南》中的[為堆疊建立快速建立連結](#)。

Note

您只能使用安裝 PCUI 所用的相同 AWS ParallelCluster 版本來建立和編輯叢集或建置映像。

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

1. 登入 AWS 管理主控台。
2. 從此處清單中選擇 AWS 區域 快速建立連結來部署 PCUI。這將帶您前往 主控台 中的 CloudFormation Create Stack Wizard。
 - [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](#)
 - [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](#)
3. 輸入管理員電子郵件的有效電子郵件地址和 ParallelCluster 版本。

部署成功完成後，PCUI 會將臨時密碼傳送至此電子郵件地址，供您用來存取 PCUI。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 PCUI。
 4. 將表單的其餘部分保留空白，或輸入（選用）參數的值來自訂 PCUI 建置。
 5. 請注意用於後續步驟的堆疊名稱。
 6. 導覽至 功能。同意 CloudFormation 功能。
 7. 選擇建立。完成 AWS ParallelCluster API 和 PCUI 部署大約需要 15 分鐘。
 8. 在建立堆疊時檢視堆疊詳細資訊。
 9. 部署完成後，開啟傳送至您輸入之地址且包含臨時密碼的管理員電子郵件。使用 存取 PCUI。（請記住，如果您在登入 PCUI 之前永久刪除電子郵件，則必須刪除您建立的 PCUI 堆疊，然後重新安裝 PCUI。
 10. 在堆疊的 CloudFormation 主控台清單中，選擇您在上一個步驟中記下的堆疊名稱連結。
 11. 在堆疊詳細資訊中，選擇輸出，然後選取名為 **Stackname** URL 的金鑰連結以開啟 PCUI（其中 **Stackname** 是您在上一個步驟中記下的名稱）。
 12. 輸入臨時密碼。依照步驟建立您自己的密碼並再次登入。

13. 您現在位於 AWS 區域 所選 中 PCUI 的首頁。
14. 若要開始使用 PCUI，請參閱 [使用 PCUI 設定和建立叢集](#)。

Note

PCUI 工作階段的預設持續時間為 5 分鐘，這是 Cognito 自 PCUI 2023.12.0 起提供的最小值。因此，預期從 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。

堆疊參數

AdditionalPoliciesPCAPI :

描述：(選用) 附加至 ParallelCluster Lambda 函數預設執行角色之其他 IAM 政策的 ARN。只能指定一個政策。

類型：字串

預設："

AllowedPattern："`^(arn : .* : iam : : .* : policy\\V(【a-zA-Z0-9_-】 +))|()$"`

AdminUserEmail :

描述：預設要設定之管理使用者的電子郵件地址 (僅適用於新的 Cognito 執行個體)。

類型：字串

預設："

CognitoCustomDomain :

描述：(選用) Cognito 的自訂網域名稱。如果省略，將使用預設的 Cognito 網域名稱。

類型：字串

預設："

CognitoCustomDomainCertificateArn :

描述：(選用) 為 Cognito 自訂網域發行之 ACM 憑證的 ARN。只有在指定時 CognitoCustomDomain，才需要此項目。

類型：字串

預設："

CustomDomain：

描述：(選用) 自訂網域名稱。如果省略，將使用預設網域名稱。

類型：字串

預設："

CustomDomainCertificateArn：

描述：(選用) 針對自訂網域發行之 ACM 憑證的 ARN。只有在指定時 CustomDomain，才需要此項目。

類型：字串

預設："

IAMRoleAndPolicyPrefix：

描述：套用至每個 IAM 角色和政策名稱的字首 (長度上限：10)。【ParallelCluster >= 3.8.0】

類型：字串

預設："

MaxLength：10

ImageBuilderSubnetId：

描述：(選用) 選取要用於建置容器映像的子網路。子網路必須是公有和自動指派IPs。如果未選取，則會使用預設子網路。

類型：字串

預設："

ImageBuilderVpcId：

描述：(選用) 選取要用於建置容器映像的 VPC。如果未選取，則會使用預設 VPC。

類型：字串

預設："

InfrastructureBucket :

描述：(選用) 存放 CloudFormation 檔案的 S3 儲存貯體。只有在測試對基礎設施本身所做的變更時，才變更此參數。

類型：字串

預設："

LambdaSecurityGroupIds :

描述：要與 PCUI Lambda 函數建立關聯的安全群組逗號分隔清單。

類型：CommaDelimitedList

預設："

LambdaSubnetIds :

描述：要與 PCUI Lambda 函數建立關聯的以逗號分隔的子網路 IDs 清單。這些子網路應為私有，並與您的 VPC 端點相關聯。

類型：CommaDelimitedList

預設："

PermissionsBoundaryPolicy :

描述：用於 ParallelCluster UI 基礎設施所建立之每個 IAM 角色的許可界限的 IAM 政策 ARN。

類型：字串

預設："

AllowedPattern："`^(arn : .* : iam : : .* : policy\\(【a-zA-Z0-9_-】 +))|()$"`

PermissionsBoundaryPolicyPCAPI :

描述：IAM 政策的 ARN，用於 ParallelCluster API 基礎設施建立的每個 IAM 角色的許可界限。

【ParallelCluster >= 3.8.0】

類型：字串

預設："

AllowedPattern："`^(arn : .* : iam : : .* : policy\\(【a-zA-Z0-9_-】 +))|()$"`

PublicEcrImageUri :

描述：指定時，ParallelCluster UI 容器 Lambda 的 Docker 映像 URI。

類型：字串

預設：public.ecr.aws/pcm/parallelcluster-ui : 2024.11.0

SNSRole :

描述：先前部署之 PCUI Cognito Stack 的 SNSRole ARN。保留空白以建立新的。

類型：字串

預設："

UserPoolAuthDomain :

描述：先前部署之 PCUI Cognito 使用者集區的 UserPoolAuthDomain。保留空白以建立新的。

類型：字串

預設："

UserPoolId :

描述：先前部署之 PCUI Cognito 使用者集區的 UserPoolId。保留空白以建立新的。

類型：字串

預設："

版本 :

描述：要部署的 AWS ParallelCluster 版本。

類型：字串

預設：3.11.1

VpcEndpointId :

描述：輸入 VPC 端點，其類型界面適用於 execute-api 服務，以啟用私有 PCUI 實作。啟用時，API 只會接受來自指定 VPC 內的請求。

類型：字串

預設："

設定自訂網域

了解如何設定 UI (PCUI) AWS ParallelCluster 的自訂網域。UI 託管於 中的 Amazon API Gateway AWS 帳戶。您可以在 API Gateway 主控台中設定自訂網域。

事前準備：

- 您有 AWS 帳戶。
- 您擁有網域。
- 網域根目錄是可解析的。例如，如果您想要使用 `xyz.example.com` 做為自訂網域，則根網域 `example.com` 必須是可解析的。這表示您至少必須有 `example.com` 指向 IP 的 A 記錄。如果您沒有此類記錄，請新增指向 `0.0.0.0` 的虛擬記錄。
- 您已在 `us-east-1` 中建立或匯入 Cognito 的自訂網域憑證至 AWS Certificate Manager (ACM)。
- 您擁有變更網域名稱系統 (DNS) 設定的必要許可。

主題

- [部署 PCUI](#)
- [設定 DNS](#)

部署 PCUI

請完成下列步驟，以建立或更新 UI AWS ParallelCluster (PCUI) 部署，使其將使用您的自訂網域。

在此範例中，我們假設您想要部署具有自訂網域的 PCUI，`xyz.example.com` 以及具有自訂網域的 Amazon Cognito 介面 `auth-xyz.example.com`。

Note

請注意，自訂 Amazon Cognito 網域並非必要項目，而且可以保留預設值。我們為完整性納入此自訂項目。

Note

不支援 Amazon Cognito 的自訂網域 AWS GovCloud (US) Regions。

若要達成此目的，請使用下列參數部署 PCUI 堆疊：

- CustomDomain : `https://xyz.example.com`。
- CustomDomainCertificateArn : `https://xyz.example.com` 的 ACM 憑證 Amazon Resource Name (ARN)。
- CognitoCustomDomain : `https://auth-xyz.example.com`。
- CognitoCustomDomainCertificateArn 至 `https://xyz.example.com` 的 ACM 憑證 Amazon Resource Name (ARN)。

設定 DNS

請完成下列步驟來設定您的網域名稱服務 (DNS)，以便 AWS ParallelCluster UI (PCUI) 和 Amazon Cognito 回應所需的自訂網域。

此範例假設您想要部署具有自訂網域的 PCUI，`xyz.example.com` 以及具有自訂網域的 Amazon Cognito 界面 `auth-xyz.example.com`。

1. 使用下列參數部署 PCUI 堆疊：

- CustomDomainEndpoint : 在您的 DNS for `xyz.example.com` 中建立一個記錄，指向輸出中指定的別名。
- CognitoCustomDomainEndpoint : 在您的 DNS for `auth-xyz.example.com` 中建立記錄，指向輸出中指定的別名。

2. 等待約 10 分鐘，以便分佈 DNS 變更。

3. 分佈 DNS 變更時，PCUI 會在 `xyz.example.com/pcui` 上回應，Amazon Cognito 身分驗證頁面會在 `auth-xyz.example.com` 上回應。

Amazon Cognito 使用者集區選項

下列各節參考 CloudFormation 快速建立連結或快速建立 URLs。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 《使用者指南》中的 [為堆疊建立快速建立連結](#)。

若要維護可與多個 UI (PCUI) 執行個體搭配使用的 Amazon Cognito AWS ParallelCluster 使用者集區，請考慮下列選項：

- 使用現有的 PCUI 執行個體，連結至從巢狀 CloudFormation 堆疊建立的 Amazon Cognito 使用者集區。這是當您使用快速建立連結部署 PCUI 並保持所有 Amazon Cognito 參數空白時所建立的。
- 首先，部署獨立的 Amazon Cognito 使用者集區。然後，部署連結至您剛部署之獨立 Amazon Cognito 使用者集區的新 PCUI 執行個體。如此一來，您可以將 Amazon Cognito 部署與 PCUI 部署分開。請注意，非巢狀 PCUI CloudFormation 堆疊更容易更新。

將現有的 Amazon Cognito 使用者集區與新的 PCUI 執行個體搭配使用

完成下列步驟，將現有的 Amazon Cognito 使用者集區與新的 PCUI 執行個體搭配使用

將現有的 Amazon Cognito 使用者集區與新的 PCUI 執行個體搭配使用

1. 在 CloudFormation 主控台中，選取包含您要與多個 PCUI 執行個體搭配使用之 Amazon Cognito 使用者集區的 PCUI 堆疊。
2. 導覽至建立 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 PCUI 執行個體，並使用您複製的輸出填入所有 External PCUI Amazon Cognito 參數。這可防止新的 PCUI 堆疊建立新的集區，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 PCUI 執行個體，並將其連結至 Amazon Cognito 使用者集區。

建立獨立的 Amazon Cognito 使用者集區

建立獨立的 Amazon Cognito 使用者集區

1. 選擇快速建立連結來啟動僅限 Amazon Cognito 的堆疊，該連結會標記您在此清單中部署 PCUI 執行個體的相同 AWS 區域 連結：
 - [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](#)
 - [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](#)
2. 堆疊建立完成後，選取輸出索引標籤並複製下列參數的值：
 - UserPoolId
 - UserPoolAuthDomain
 - SNSRole
 3. 選擇 PCUI 快速入門連結並使用您複製的值填入所有 External PCUI Amazon Cognito 參數，以部署新的 PCUI 執行個體。新的 PCUI 執行個體會連結至獨立的 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 PCUI 執行個體，並將其連結至獨立的 Amazon Cognito 使用者集區。

識別 AWS ParallelCluster 和 PCUI 版本

若要識別 AWS ParallelCluster 和 PCUI 版本：

1. 在 CloudFormation 主控台中，選取 PCUI 堆疊。
2. 選取參數索引標籤。
3. ~~AWS ParallelCluster 版本是參數版本的值。~~

- PCUI 版本位於 PublicEcrImageUri 值的結尾。例如，如果值為 `public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02`，則版本為 `2023.02`。

Note

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

PCUI 成本

PCUI 是以無伺服器架構為基礎，您可以在免費 AWS 方案類別中使用它。下表列出 PCUI 所依賴 AWS 服務的及其自由方案限制。一般用量估計每月成本低於 1 美元。

服務	AWS 免費方案
Amazon Cognito	每月 50,000 個作用中使用者
Amazon API Gateway	100 萬次靜態 API 呼叫
AWS Lambda	每月 100 萬個免費請求和每月 400,000 GB 的運算時間
EC2 Image Builder	除了 EC2 之外，免費
Amazon Elastic Compute Cloud	15 分鐘的一次性容器映像建置
CloudFormation	5 GB 資料 (擷取、封存儲存和由 Logs Insights 查詢掃描的資料)

入門 AWS ParallelCluster

若要開始使用命令列界面 (CLI) 或 Web 型使用者介面 (UI) AWS ParallelCluster 來 AWS ParallelCluster 設定和建立叢集。PCUI 已在 3.5.0 版中新增。

主題

- [使用 AWS ParallelCluster 命令列界面設定和建立叢集](#)
- [使用 PCUI 設定和建立叢集](#)

- [連接至叢集](#)

使用 AWS ParallelCluster 命令列界面設定和建立叢集

安裝之後 AWS ParallelCluster，請完成下列組態步驟。

1. 確認 AWS 您的帳戶具有的角色包含執行 CLI [pcluster](#) 所需的許可。如需詳細資訊，請參閱 [AWS ParallelCluster 範例 pcluster 使用者政策](#)。
2. 設定您的 AWS 登入資料。如需詳細資訊，請參閱《AWS CLI 使用者指南》中的 [設定 AWS CLI](#)。

```
$ 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]:
```

3. 啟動 AWS 區域 叢集的 必須至少有一個 Amazon EC2 金鑰對。如需詳細資訊，請參閱《[Amazon Elastic Compute Cloud Linux 執行個體使用者指南](#)》中的 [Amazon Elastic Compute Cloud 金鑰對](#)。

當您使用 AWS ParallelCluster 命令列界面 (CLI) 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱 [AWS 使用的 服務 AWS ParallelCluster](#)。

設定和建立您的第一個叢集

若要建立您的第一個叢集，請使用 `pcluster configure` CLI 命令啟動精靈，提示您輸入設定和建立叢集所需的所有資訊。相較於使用 `slurm`，使用 AWS Batch 做為排程器時，序列的詳細資訊會有所不同。

Slurm

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

從有效 AWS 區域 識別符清單中，選擇您要叢集執行 AWS 區域 的位置。

Note

AWS 區域 顯示的 清單是以您帳戶的分割區為基礎，且僅包含為您的帳戶啟用 AWS 區域 的。如需 AWS 區域 為您的帳戶啟用的詳細資訊，請參閱《》中的 [管理 AWS 區域](#)。

一般參考。顯示的範例來自 AWS 全域分割區。如果您的帳戶位於 AWS GovCloud (US) 分割區中，則只會列出該分割區 AWS 區域中的 (gov-us-east-1 和 gov-us-west-1)。同樣地，如果您的帳戶位於 AWS 中國分割區中，則只會顯示 cn-northwest-1 cn-north-1 和 。如需支援的完整清單 AWS 區域 AWS ParallelCluster，請參閱 [支援 AWS 區域 AWS ParallelCluster](#)。

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. ap-southeast-3
9. ap-southeast-5
10. ap-southeast-7
11. ca-central-1
12. eu-central-1
13. eu-north-1
14. eu-south-1
15. eu-west-1
16. eu-west-2
17. eu-west-3
18. il-central-1
19. me-south-1
20. sa-east-1
21. us-east-1
22. us-east-2
23. us-west-1
24. us-west-2

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

從在選取的 Amazon Elastic Compute Cloud 註冊的金鑰對中選擇金鑰對 AWS 區域：

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. alinux2023
3. ubuntu2404
4. ubuntu2204
6. rhel8
7. rhel9
Operating System [alinux2]:
```

選擇前端節點執行個體類型：

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

選擇佇列組態。注意：無法為相同佇列中的多個運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]:
```

啟用 EFA 以執行需要大規模進行高階執行個體間通訊的應用程式 AWS ，無需額外費用：

- 選擇 [支援 Elastic Fabric Adapter \(EFA\)](#) 的執行個體類型。
- 啟用 [EFA](#)。
- 指定現有的 [置放群組](#) 名稱。如果您將其保留空白，會為您 AWS ParallelCluster 建立一個。

```
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，決定要使用現有的 VPC 或讓 AWS ParallelCluster 為您建立 VPC。如果您沒有正確設定的 VPC，AWS ParallelCluster 可以為您建立新的 VPC。它會同時將前端和運算節點放在相同的公有子網路中，或只將前端節點放在私有子網路中具有所有運算節點的公有子網路中。如果您讓 AWS ParallelCluster 建立 VPC，則必須決定所有節點是否位於公有子網路中。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](#)以取得其他聯網需求。

您可以達到中允許的 VPCs 數量配額 AWS 區域。預設配額為 VPCs AWS 區域。如需此配額以及如何請求增加的詳細資訊，請參閱《Amazon [VPC 使用者指南](#)》中的 [VPC 和子網路](#)。

Important

根據預設，建立的 VPCs AWS ParallelCluster 不會啟用 VPC 流程日誌。VPC 流程日誌可讓您擷取往返 VPCs 中網路介面之 IP 流量的相關資訊。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 [VPC 流程日誌](#)。

如果您讓 AWS ParallelCluster 建立 VPC，請確定您決定所有節點是否位於公有子網路中。

Note

如果您選擇 1. Head node in a public subnet and compute fleet in a private subnet，會 AWS ParallelCluster 建立會產生額外費用的 NAT 閘道，即使您指定免費方案資源也一樣。

```
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

```

如果您未建立新的 VPC，則必須選取現有的 VPC。

如果您選擇讓 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 cluster-config.yaml
```

從有效 AWS 區域 識別符清單中，選擇您要叢集執行 AWS 區域 的位置。

Note

AWS 區域 顯示的 清單取決於您帳戶的分割區。它只包含為您的帳戶 AWS 區域 啟用的。如需 AWS 區域 為您的帳戶啟用的詳細資訊，請參閱《》中的 [管理 AWS 區域](#)。AWS 一般參考。顯示的範例來自 AWS 全域分割區。如果您的帳戶位於 AWS GovCloud (US) 分割區中，則只會列出該分割區 AWS 區域 中的 (gov-us-east-1 和 gov-us-west-1)。同樣地，如果您的帳戶位於 AWS 中國分割區中，則只會顯示 cn-northwest-1 和 cn-

north-1和。如需 AWS 區域 支援的完整清單 AWS ParallelCluster，請參閱 [支援 AWS 區域 AWS ParallelCluster](#)。

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. ap-southeast-3
9. ap-southeast-5
10. ap-southeast-7
11. ca-central-1
12. eu-central-1
13. eu-north-1
14. eu-south-1
15. eu-west-1
16. eu-west-2
17. eu-west-3
18. il-central-1
19. me-south-1
20. sa-east-1
21. us-east-1
22. us-east-2
23. us-west-1
24. us-west-2

AWS ## ID [us-east-1]:

金鑰對是從所選 中向 Amazon EC2 註冊的金鑰對中選取 AWS 區域。選擇金鑰對：

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。輸入前端節點執行個體類型：

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

選擇佇列組態。AWS Batch 排程器只包含單一佇列。已輸入運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 VPCs 或讓 AWS ParallelCluster 為您建立 VPCs。如果您沒有適當設定的 VPC，AWS ParallelCluster 可以建立一個新的 VPC。它在相同的公有子網路中同時使用前端節點和運算節點，或僅使用公有子網路中具有私有子網路中所有節點的前端節點。您可以達到區域中允許的 VPCs 數量配額。VPC 的預設數目為五。如需此配額以及如何請求增加的詳細資訊，請參閱《Amazon [VPC 使用者指南](#)》中的 [VPC 和子網路](#)。

Important

根據預設，建立的 VPCs AWS ParallelCluster 不會啟用 VPC 流程日誌。VPC 流程日誌可讓您擷取往返 VPCs 中網路介面之 IP 流量的相關資訊。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 [VPC 流程日誌](#)。

如果您讓 AWS ParallelCluster 建立 VPC，請確定您決定所有節點是否位於公有子網路中。

Note

如果您選擇 1. Head node in a public subnet and compute fleet in a private subnet，AWS ParallelCluster 會建立會產生額外費用的 NAT 閘道，即使您指定免費方案資源也一樣。

```
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
```

如果您未建立新的 VPC，則必須選取現有的 VPC。

如果您選擇讓 AWS ParallelCluster 建立 VPC，請記下 VPC ID，以便您稍後使用 AWS CLI 或 AWS 管理主控台 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，請務必決定是否使用現有的子網路或建立新的子網路。

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

```
Creating CloudFormation stack...
```

```
Do not leave the terminal until the process has finished
```

當您完成上述步驟時，簡單的叢集會啟動至 VPC。VPC 使用支援公有 IP 地址的現有子網路。子網路的路由表為 `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 也必須具有 DHCP 選項，且 domain-name 適用於 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 閘道或內部代理來啟用運算節點的 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": "3.15.0",  
    "clusterStatus": "CREATE_IN_PROGRESS"  
  },  
  "validationMessages": []  
}
```

遵循叢集進度：

```
$ 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 用戶端設定來連線到叢集。如需連線至 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，您可以透過 [連接叢集](#)

```
$ 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 CloudFormation 網路堆疊來刪除 VPC 中的網路資源。堆疊的名稱開頭為「parallelclusternetworking-」，並包含「YYYYMMDDHHMMSS」格式的建立時間。您可以使用 [list-stacks](#) 命令列出堆疊。

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

您可以使用 [delete-stack](#) 命令刪除堆疊。

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

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

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

使用 PCUI 設定和建立叢集

PCUI 是以 Web 為基礎的使用者介面，可鏡射 pcluster AWS ParallelCluster CLI，同時提供類似主控台的體驗。您可以在 中安裝和存取 PCUI AWS 帳戶。當您執行時，PCUI 會存取 中 Amazon AWS ParallelCluster API Gateway 上託管的 API 執行個體 AWS 帳戶。Amazon API Gateway

Note

PCUI 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案，或使用 AWS ParallelCluster CLI。

本節說明如何使用 PCUI 設定和建立叢集。

事前準備：

- 存取執行中的 PCUI 執行個體。如需詳細資訊，請參閱 [安裝 PCUI](#)。

設定和建立叢集

1. 在 PCUI 叢集檢視中，依序選擇建立叢集。
2. 在叢集名稱中，輸入叢集的名稱。
3. 為您的叢集選擇具有公有子網路的 VPC，然後選擇下一步。
4. 在標頭節點中，選擇新增 SSM 工作階段，然後選擇下一步。
5. 在佇列、運算資源中，為靜態節點選擇 1。
6. 針對執行個體類型，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. 在儲存中，選擇下一步。
8. 在叢集組態中，檢閱叢集組態 YAML，然後選擇乾執行進行驗證。
9. 根據已驗證的組態，選擇建立以建立叢集。
10. 幾秒鐘後，PCUI 會自動將您導覽回叢集，您可以在其中監控叢集建立狀態和堆疊事件。
11. 選擇詳細資訊以查看叢集詳細資訊，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 AWS 管理主控台 建立叢集的 CloudFormation 堆疊連結。
14. 在詳細資訊中，叢集建立完成後，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，請選擇 Shell 以存取叢集前端節點。

Note

當您選擇 Shell 時，AWS ParallelCluster 請開啟 Amazon EC2 Systems Manager 工作階段並將新增至 `ssm-user/etc/sudoers`。如需詳細資訊，請參閱《Amazon EC2 Systems Manager 使用者指南》中的[開啟或關閉 ssm-user 帳戶管理許可](#)。

16. 若要清除，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，您可以連線至叢集前端節點來執行任務、檢視結果、管理使用者，以及監控叢集和任務狀態。若要連線至叢集前端節點執行個體，請使用下列其中一種方法：

- 您可以 ssh 搭配 [金鑰對](#) 使用 來登入。在叢集組態的 [HeadNode / KeyName](#) 中指定私有金鑰。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](#)。

- 您可以使用 `pcluster ssh` 命令列界面 (CLI) 命令來登入。在叢集組態 [HeadNode/](#) 中指定私有金鑰 [KeyName](#)。如需詳細資訊，請參閱 [pcluster ssh](#)。
- 您可以使用 SSM 工作階段連線到叢集前端節點。您必須將 `AmazonSSMManagedInstanceCore` 受管政策新增至叢集組態 [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](#)。
- 您可以使用 Amazon DCV 連線到叢集前端節點。如需詳細資訊，請參閱 [透過 Amazon DCV 連接至前端和登入節點](#)。
- 使用 PCUI 時，您也可以使用 UI 提供的 Amazon EC2 Connect 命令來連線至叢集前端節點。

多個使用者存取叢集

了解如何實作和管理多個使用者對單一叢集的存取。

在本主題中，AWS ParallelCluster 使用者是指運算執行個體的系統使用者。範例是 Amazon EC2 執行個體 `ec2-user` 的。

AWS ParallelCluster 目前可使用的 [所有](#) 都提供多使用者存取支援 AWS 區域 AWS ParallelCluster。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 [AWS ParallelCluster DirectoryService configuration](#)。AWS Directory Service directories 可以連接到多個叢集。這允許跨多個環境集中管理身分，並提供統一的登入體驗。

當您使用 AWS Directory Service 進行 AWS ParallelCluster 多個使用者存取時，您可以使用目錄中定義的使用者登入資料登入叢集。這些登入資料由使用者名稱和密碼組成。第一次登入叢集後，系統會自動產生使用者 SSH 金鑰。您可以使用它來登入，無需密碼。

在部署目錄服務之後，您可以建立、刪除和修改叢集的使用者或群組。您可以使用 AWS Directory Service 或使用 Active Directory 使用者和電腦工具，在 AWS 管理主控台中執行此操作。此工具可從加入 Active Directory 的任何 Amazon EC2 執行個體存取。如需詳細資訊，請參閱 [安裝 Active Directory 管理工具](#)。

如果您打算在沒有網際網路存取的單一子網路 AWS ParallelCluster 中使用，請參閱 [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#) 以取得其他需求。

主題

- [建立 Active Directory](#)

- [建立具有 AD 網域的叢集](#)
- [登入與 AD 網域整合的叢集](#)
- [執行 MPI 任務](#)
- [AWS Managed Microsoft AD 透過 LDAP\(S\) 叢集組態的範例](#)

建立 Active Directory

請務必先建立 Active Directory (AD)，再建立叢集。如需有關如何為叢集選擇作用中目錄類型的資訊，請參閱《AWS Directory Service 管理指南》中的[選擇項目](#)。

如果目錄是空的，請新增具有使用者名稱和密碼的使用者。如需詳細資訊，請參閱 [AWS Directory Service for Microsoft Active Directory](#) 或 [Simple AD](#) 特定的文件。

Note

AWS ParallelCluster 要求每個 Active Directory 使用者目錄都位於 `/home/$user` 目錄中。

建立具有 AD 網域的叢集

Warning

此簡介章節說明如何透過輕量型目錄存取通訊協定 (LDAP) AWS ParallelCluster 使用 Managed Active Directory (AD) 伺服器進行設定。LDAP 是不安全的通訊協定。對於生產系統，我們強烈建議使用 TLS 憑證 (LDAPS)，如以下[AWS Managed Microsoft AD 透過 LDAP\(S\) 叢集組態的範例](#)章節所述。

透過在叢集組態檔案的 `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，AWS Managed Microsoft AD 透過輕量型目錄存取通訊協定 (LDAP) 整合叢集與。

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

- 您必須將 `ldap_auth_disable_tls_never_use_in_production` 參數設定為 [DirectoryService](#) / `True` 下 [AdditionalSssdConfigs](#)。

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

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

的叢集組態 AWS Managed Microsoft AD：

```
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
```

若要將此組態用於 Simple AD，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：

- 我們建議您透過 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](#) 屬性值符合 `DnsIpAddrsdescribe-directories` 輸出清單中的項目。
- 建議您的叢集使用位於 [DirectoryService DomainAddr](#) 指向之相同可用區域中的子網路。如果您使用建議用於目錄 VPCs 的 [自訂動態主機組態通訊協定 \(DHCP\) 組態](#)，且子網路不位於 [DirectoryService](#) / [DomainAddr](#) 可用區域中，則可以跨可用區域進行跨流量。使用多使用者 AD 整合功能不需要使用自訂 DHCP 組態。
- [DirectoryService](#) / [DomainReadOnlyUser](#) 屬性值指定必須在目錄中建立的使用者。根據預設，不會建立此使用者。建議您不要授予此使用者修改目錄資料的許可。
- [PasswordSecretArn](#) 屬性值指向 AWS Secrets Manager 秘密，其中包含您為 [DirectoryService DirectoryService/ DomainReadOnlyUser](#) 屬性指定的使用者密碼。如果此使用者的密碼變更，請更新秘密值並更新叢集。若要更新叢集的新秘密值，您必須使用 `pcluster update-compute-fleet` 命令停止運算機群。如果您將叢集設定為使用 [LoginNodes](#)，請停止 [LoginNodes](#) / [Pools](#) 並在將 [Pools](#) / [LoginNodes](#) 設定為 [Count](#) 0 之後更新叢集。然後，從叢集前端節點內執行下列命令。

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

如需另一個範例，另請參閱 [整合 Active Directory](#)。

登入與 AD 網域整合的叢集

如果您啟用 Active Directory (AD) 網域整合功能，則會在叢集前端節點上啟用密碼身分驗證。AD 使用者的主目錄會在第一次使用者登入前端節點時，或第一次 `sudo-user` 切換到前端節點上的 AD 使用者時建立。

叢集運算節點未啟用密碼身分驗證。AD 使用者必須使用 SSH 金鑰登入以運算節點。

根據預設，SSH 金鑰會在第一次登入前端節點的 SSH 時，於 AD 使用者/`${HOME}`/.ssh 目錄中設定。此行為可以透過在叢集組態 `false` 中將 [DirectoryService / GenerateSshKeysForUsers](#) 布林值屬性設定為 `來` 停用。根據預設，[DirectoryService GenerateSshKeysForUsers](#) 設定為 `true`。

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

AWS Managed Microsoft AD 密碼會在 42 天後過期。如需詳細資訊，請參閱《Directory Service 管理指南》中的[管理的密碼政策 AWS Managed Microsoft AD](#)。如果密碼過期，則必須重設以還原叢集存取。如需詳細資訊，請參閱[如何重設使用者密碼和過期的密碼](#)。

Note

如果 AD 整合功能無法如預期運作，SSSD 日誌可以提供有用的診斷資訊，以對問題進行故障診斷。這些日誌位於叢集節點的 `/var/log/sss` 目錄中。根據預設，它們也會存放在叢集的 Amazon CloudWatch 日誌群組中。

如需詳細資訊，請參閱[對與 Active Directory 的多使用者整合進行故障診斷](#)。

執行 MPI 任務

如 SchedMD 中所述，使用 Slurm 作為 MPI 引導方法引導 MPI 任務。如需詳細資訊，請參閱 MPI 程式庫的官方[Slurm 文件](#)或官方文件。

例如，在 [IntelMPI 官方文件中](#)，您會了解在執行 StarCCM 任務時，必須透過匯出環境變數，將設定為 Slurm 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 作為產生 MPI 任務的機制，則可能發生 [Slurm 中的已知錯誤](#)，導致錯誤解析目錄使用者名稱為「沒有人」。

您可以將應用程式設定為使用 Slurm 做為 MPI 引導方法，或參閱疑難排解一節[使用者名稱解析的已知問題](#)中的以取得更多詳細資訊和可能的解決方法。

AWS Managed Microsoft AD 透過 LDAP(S) 叢集組態的範例

AWS ParallelCluster 透過輕量型目錄存取通訊協定 (LDAP) 或 LDAP over TLS/SSL (LDAPS) 與整合 AWS Directory Service，支援多個使用者存取。

下列範例示範如何建立叢集組態，以透過 AWS Managed Microsoft AD LDAP(S) 與整合。

AWS Managed Microsoft AD 透過 LDAPS 進行憑證驗證

您可以使用此範例來整合您的叢集與 AWS Managed Microsoft AD 透過 LDAPS 的，以及憑證驗證。

具有憑證組態之 AWS Managed Microsoft AD 透過 LDAPS 的 的特定定義：

- [DirectoryService](#) 對於具有憑證驗證的 LDAPS，/[LdapTlsReqCert](#) 必須設定為 hard (預設)。
- [DirectoryService](#) / [LdapTlsCaCert](#) 必須指定授權憑證 (CA) 憑證的路徑。

CA 憑證是一種憑證套件，其中包含為 AD 網域控制站發行憑證的整個 CA 鏈的憑證。

您的 CA 憑證和憑證必須安裝在叢集節點上。

- 必須指定 / [DirectoryService](#) 的控制器主機名稱 [DomainAddr](#)，而非 IP 地址。
- [DirectoryService](#) / [DomainReadOnlyUser](#) 語法必須如下所示：

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

透過 LDAPS 使用 AD 的範例叢集組態檔案：

```
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://&example-s3-bucket;/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://&example-s3-bucket;/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和設定網域控制站：

```

*#!/bin/bash*
set -e

AD_CERTIFICATE_S3_URI="s3://amzn-s3-demo-bucket/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 Windows 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
```

從 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 無需憑證驗證即可透過 LDAPS

您可以使用此範例來整合叢集與 AWS Managed Microsoft AD 透過 LDAPS 的，無需憑證驗證。

沒有憑證驗證組態的 AWS Managed Microsoft AD 透過 LDAPS 之的特定定義：

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

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

無需憑證驗證即可 AWS Managed Microsoft AD 透過 LDAPS 使用的範例叢集組態檔案：

```
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

以下各節提供使用的最佳實務 AWS ParallelCluster，其中包括網路效能和預算提醒。即使您遵循這些最佳實務仍遇到問題，請參閱 [AWS ParallelCluster 故障診斷](#) 以取得可能的解決方案。

最佳實務：前端節點執行個體類型選取

即使前端節點未執行任務，其函數及其大小對於叢集的整體效能至關重要。當您選擇要用於前端節點的執行個體類型時，請考慮下列特性：

叢集大小：前端節點會協調叢集的擴展邏輯，並負責將新節點連接至排程器。若要擴展和縮減具有大量節點的叢集，請為前端節點提供一些額外的運算容量。

共用檔案系統：當您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。確保前端節點能夠為叢集公開足夠的 NFS 伺服器目錄，並處理運算節點和前端節點之間需要共用的成品。

最佳實務：網路效能

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

- 置放群組：如果您使用的是 Slurm，請考慮設定每個Slurm佇列以使用叢集置放群組。叢集的置放群組是單一可用區域內執行個體的邏輯分組。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[置放群組](#)。您可以在佇列的 [Networking](#) 區段 [PlacementGroup](#) 中指定，每個運算資源都會指派給佇列的置放群組。在運算資源的 [Networking](#) 區段 [PlacementGroup](#) 中指定時，該特定運算資源會指派給該置放群組。運算資源置放群組規格會覆寫運算資源的佇列規格。如需詳細資訊，請參閱 [SlurmQueues / Networking / PlacementGroup](#) 和 [SlurmQueues / ComputeResources / Networking / PlacementGroup](#)。

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

或者，請為您 AWS ParallelCluster 建立置放群組。

```
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。當您在佇列中指定要啟用的置放群組，而沒有 name 或 Id 時，每個運算資源都會指派自己的受管置放群組，而不是整個佇列的一個受管群組。這有助於減少容量不足的錯誤。如果您需要一個置放群組用於整個佇列，您可以使用具名置放群組。

[SlurmQueues / Networking / PlacementGroup / Name](#) 已新增為 [SlurmQueues / Networking / PlacementGroup /](#) 的偏好替代方案 [Id](#)。

如需詳細資訊，請參閱[Networking](#)。

- 增強型聯網：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[目前世代的執行個體](#)。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](#)。
- Elastic Fabric Adapter：若要支援高階可擴展的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) 繞過硬體可增強執行個體與隨需彈性和彈性的執行個體通訊 AWS 雲端。您可以設定每個Slurm佇列[ComputeResource](#)使用 [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 EFA 的詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的 [Elastic Fabric Adapter](#)。

- 執行個體頻寬：頻寬會隨執行個體大小而擴展。如需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 最佳化執行個體](#) Amazon EC2 [EBS 磁碟區類型](#)。

最佳實務：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 資源建立定義的預算閾值警示。如需詳細資訊，請參閱 AWS Budgets 《使用者指南》中的 [設定預算動作](#)。同樣地，您也可以使用 Amazon CloudWatch 建立帳單警示。如需詳細資訊，請參閱 [建立帳單警示來監控預估的 AWS 費用](#)。

最佳實務：將叢集移至新的 AWS ParallelCluster 次要或修補程式版本

目前每個 AWS ParallelCluster 次要版本都與其 pcluster CLI 一起獨立。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：

- 在叢集外部建立的外部磁碟區中儲存個人資料，例如 Amazon EFS 和 FSx for Lustre。透過這樣做，您可以在未來輕鬆地將資料從一個叢集移至另一個叢集。
- 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 管理主控台。
 - [SharedStorage](#) / [EbsSettings](#) / [VolumeId](#)
 - [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)
 - [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)

將叢集組態中的檔案系統或磁碟區定義為現有的檔案系統或磁碟區。如此一來，它們會在您刪除叢集時保留，並且可以連接到新的叢集。

我們建議您使用 Amazon EFS 或 FSx for Lustre 檔案系統。這兩個系統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，將這些系統連接到新的叢集。

- 使用 [自訂引導操作](#) 來自訂執行個體，而不是使用自訂 AMI。如果改為使用自訂 AMI，則需要為每個新版本版本刪除並重新建立該 AMI。
- 建議您以下列順序套用上述建議：
 1. 更新現有的叢集組態，以使用現有的檔案系統定義。

2. 驗證pcluster版本並視需要更新。
3. 建立和測試新叢集。當您測試新叢集時，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. 新叢集經過完整測試並運作，且不再需要現有的叢集後，請將其刪除。

從 AWS ParallelCluster 2.x 移至 3.x

下列各節說明從 AWS ParallelCluster 2.x 移至 3.x 時會發生的情況，包括從一個版本到另一個版本的變更。

自訂引導操作

透過 AWS ParallelCluster 3，您可以使用 [和 Scheduling/ SlurmQueues](#) 區段中的 `OnNodeStart(post_install第 2 AWS ParallelCluster 版中的)` 和 `OnNodeConfigured(post_install第 2 AWS ParallelCluster 版中的)` 參數，為前端節點 [HeadNode](#) 和運算節點指定不同的自訂引導操作指令碼。如需詳細資訊，請參閱 [自訂引導操作](#)。

為 AWS ParallelCluster 2 開發的自訂引導操作指令碼必須調整為可用於 AWS ParallelCluster 3：

- 我們不建議使用 `/etc/parallelcluster/cfnconfig` 和 `cfn_node_type` 來區分頭部和運算節點。反之，我們建議您在 [HeadNode](#) 和 [Scheduling / SlurmQueues](#) 中指定兩個不同的指令碼。
- 如果您想要繼續載入 `/etc/parallelcluster/cfnconfig` 以供引導操作指令碼使用，請注意的值已從 "MasterServer" `cfn_node_type` 變更為 "HeadNode" (請參閱：[包容性語言](#))。
- 在 AWS ParallelCluster 2 上，引導操作指令碼的第一個輸入引數是指令碼的 S3 URL，並已保留。在 AWS ParallelCluster 3 中，只有組態中設定的引數會傳遞至指令碼。

Warning

正式不支援使用透過 `/etc/parallelcluster/cfnconfig` 檔案提供的內部變數。在未來版本中，可能會移除此檔案。

AWS ParallelCluster 2.x 和 3.x 使用不同的組態檔案語法

AWS ParallelCluster 3.x 組態使用 YAML 語法。您可以在 [找到完整的參考組態檔案](#)。

除了需要 YAML 檔案格式之外，AWS ParallelCluster 3.x 中也已更新許多組態區段、設定和參數值。在本節中，我們會記下 AWS ParallelCluster 組態的重要變更，以及說明每個版本之間這些差異的 side-by-side 範例 AWS ParallelCluster。

啟用和停用超執行緒的多個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

```

新的 FSx for Lustre 檔案系統組態範例

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://amzn-s3-demo-bucket
export_path = s3://amzn-s3-demo-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://amzn-s3-demo-bucket
      ExportPath: s3://amzn-s3-demo-bucket/export_dir
      WeeklyMaintenanceStartTime: "3:02:30"
      DeploymentType: PERSISTENT_1
      DataCompressionType: LZ4

```

掛載現有 FSx for Lustre 檔案系統的叢集組態範例

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
```

具有 Intel HPC 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受適用 [Intel 最終使用者授權合約](#) 的條款和條件的約束。

自訂 IAM 組態範例，包括：執行個體描述檔、執行個體角色、執行個體的其他政策，以及與叢集相關聯的 lambda 函數角色

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 , `additional_iam_policies` 且無法與 `ec2_iam_role` 搭配使用。
- 對於 AWS ParallelCluster 3 , 您可以對頭部和運算節點有不同的 IAM 設定 , 甚至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 執行個體描述檔做為 IAM 角色的替代方案。 `InstanceProfileInstanceRole` 或 `AdditionalIamPolicies` 無法一起設定。

自訂引導操作的範例

AWS ParallelCluster 2 :

```
[cluster default]
s3_read_resource = arn:aws:s3:::amzn-s3-demo-bucket/*
pre_install = s3://amzn-s3-demo-bucket/scripts/pre_install.sh
pre_install_args = 'R curl wget'
post_install = s3://amzn-s3-demo-bucket/scripts/post_install.sh
post_install_args = "R curl wget"
...
```

AWS ParallelCluster 3 :

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

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:::amzn-s3-demo-bucket/read_only/*
s3_read_write_resource = arn:aws:s3:::amzn-s3-demo-bucket/read_and_write/*
...
```

AWS ParallelCluster 3 :

```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: amzn-s3-demo-bucket
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: amzn-s3-demo-bucket
        KeyName: read_and_write/
        EnableWriteAccess: True
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue1
      ...
    Iam:
      S3Access:
        - BucketName: amzn-s3-demo-bucket
          KeyName: read_only/
          EnableWriteAccess: False
        - BucketName: amzn-s3-demo-bucket
          KeyName: read_and_write/
          EnableWriteAccess: True
```

包容性語言

AWS ParallelCluster 3 在「主要」使用的位置使用「主節點」一詞 AWS ParallelCluster 2。這包含下列項目：

- 在 AWS Batch 任務環境中匯出的變數已變更：從 MASTER_IP 變更為 PCLUSTER_HEAD_NODE_IP。
- 所有 CloudFormation 輸出從 變更為 Master* HeadNode*。
- 所有 NodeType 和標籤從 變更為 Master HeadNode。

排程器支援

AWS ParallelCluster 3.x 不支援網格引擎 (SGE) 和扭力排程器。

AWS Batch 命令 `awsbhosts`、`awsbkill`、`awsbout`、`awsbstat`、`awsbqueues` 和 `awsbsub` 會以單獨的 `aws-parallelcluster-awsbatch-cli` PyPI 套件形式發佈。此套件由安裝在前端節點 AWS ParallelCluster 上。您仍然可以從叢集的前端節點使用這些 AWS Batch 命令。不過，如果您想要從前端節點以外的位置使用 AWS Batch 命令，您必須先安裝 `aws-parallelcluster-awsbatch-cli` PyPI 套件。

AWS ParallelCluster CLI

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

設定新的叢集

相較於 AWS ParallelCluster 2，`pcluster configure` 命令包含不同的參數 in AWS ParallelCluster 3。如需詳細資訊，請參閱 [pcluster configure](#)。

另請注意，組態檔案語法已從 AWS ParallelCluster 2 變更。如需叢集組態設定的完整參考，請參閱 [叢集組態檔案](#)。

建立新的叢集

AWS ParallelCluster `pcluster create` 命令已取代 2 的 [pcluster create-cluster](#) 命令。

請注意，在沒有 `-nw` 選項的情況下，in AWS ParallelCluster 2.x 的預設行為是等待叢集建立事件，而 AWS ParallelCluster 3.x 命令會立即傳回。您可以使用 監控叢集建立的進度 [pcluster describe-cluster](#)。

An AWS ParallelCluster 3 組態檔案包含單一叢集定義，因此不再需要 `-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 命令必須以 [pcluster list-clusters](#) 命令取代。

注意：您需要 AWS ParallelCluster v2 CLI 來列出使用 2 AWS ParallelCluster.x 版本 建立的叢集。如需如何使用 AWS ParallelCluster 虛擬環境安裝多個版本的 [AWS ParallelCluster 在虛擬環境中安裝 \(建議\)](#)，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 命令必須以 [pcluster update-compute-fleet](#) 命令取代。

啟動運算機群：

```
# 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

```

停止運算機群：

```

# 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 命令在 AWS ParallelCluster 3.x 中具有不同的參數名稱。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 組態更新

從 3.0.0 版開始，根據預設，AWS ParallelCluster 引入了將對前端節點 IMDS（和執行個體描述檔登入資料）的存取限制為超級使用者子集的支援。如需詳細資訊，請參閱[Imds 屬性](#)。

使用 AWS ParallelCluster

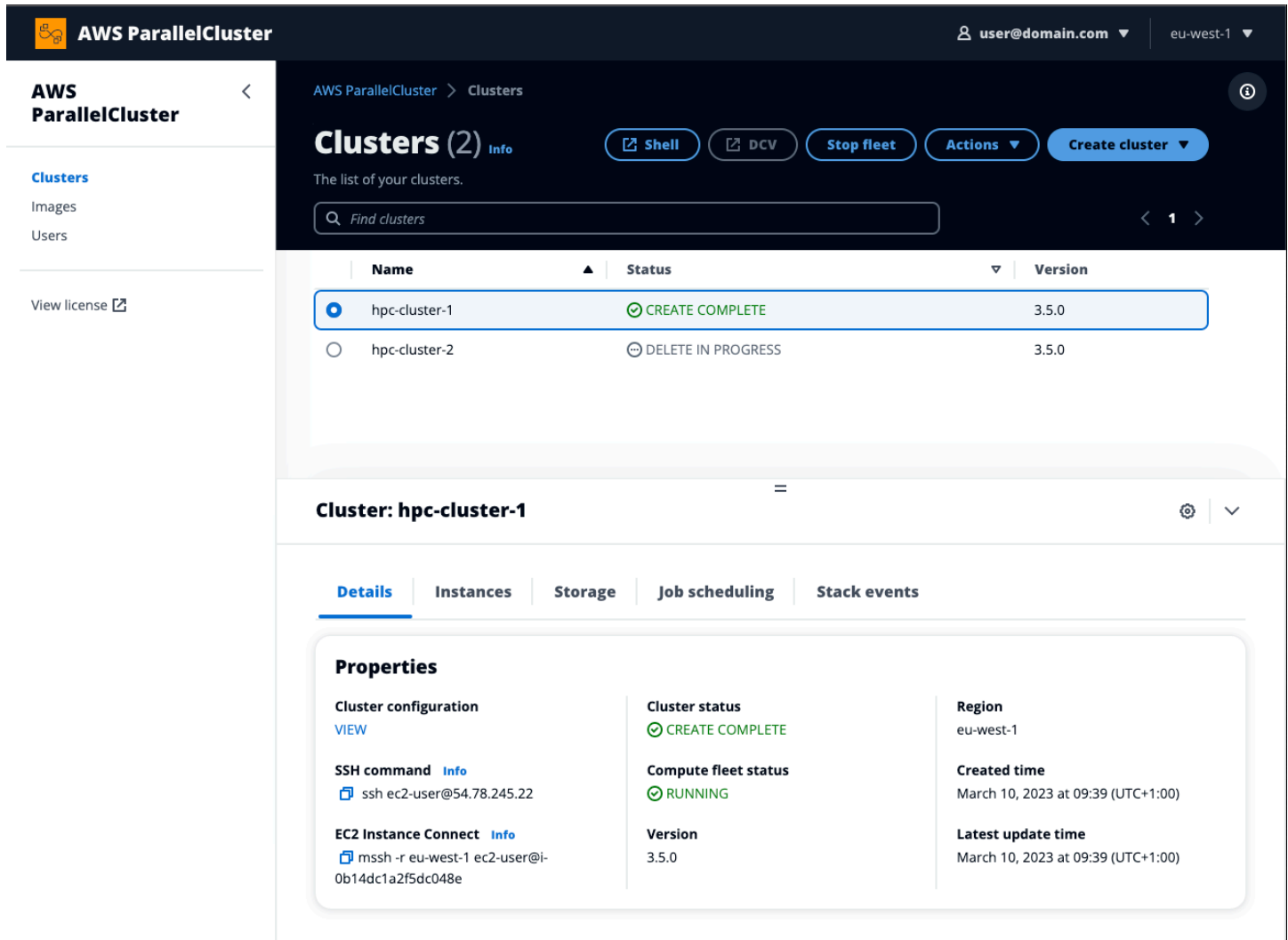
主題

- [AWS ParallelCluster UI](#)
- [AWS Lambda 中的 VPC 組態 AWS ParallelCluster](#)
- [AWS Identity and Access Management 中的 許可 AWS ParallelCluster](#)
- [網路組態](#)
- [佈建的登入節點 AWS ParallelCluster](#)
- [自訂引導操作](#)
- [使用 Amazon S3](#)
- [使用 競價型執行個體](#)
- [支援的排程器 AWS ParallelCluster](#)
- [共用儲存](#)
- [AWS ParallelCluster 資源和標記](#)
- [監控 AWS ParallelCluster 和日誌](#)
- [AWS CloudFormation 自訂資源](#)
- [Elastic Fabric Adapter](#)
- [啟用 Intel MPI](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster 適用於 Terraform](#)
- [透過 Amazon DCV 連接至前端和登入節點](#)
- [使用 pcluster update-cluster](#)
- [AWS ParallelCluster AMI 自訂](#)
- [使用隨需容量預留 \(ODCR\) 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 執行個體替換](#)
- [作業系統](#)

AWS ParallelCluster UI

UI AWS ParallelCluster (PCUI) 是以 Web 為基礎的使用者介面，可做為儀表板來建立、監控和管理叢集。您可以在 中安裝和存取 PCUI AWS 帳戶。PCUI 會以 3.5.0 AWS ParallelCluster 版新增。

若要安裝 PCUI 並開始使用，請參閱 [安裝 PCUI](#) 和 [使用 PCUI 設定和建立叢集](#)。



The screenshot displays the AWS ParallelCluster UI interface. At the top, the header shows 'AWS ParallelCluster' and the user 'user@domain.com' in the 'eu-west-1' region. The main navigation menu on the left includes 'Clusters', 'Images', and 'Users'. The central area shows a 'Clusters (2)' overview with buttons for 'Shell', 'DCV', 'Stop fleet', 'Actions', and 'Create cluster'. Below this is a search bar and a table of clusters:

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

The 'hpc-cluster-1' cluster is selected, and its details are shown below. The 'Details' tab is active, displaying the following properties:

- Cluster configuration:** [VIEW](#)
- SSH command:** `ssh ec2-user@54.78.245.22`
- EC2 Instance Connect:** `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)

PCUI 支援下列功能：

- 顯示下列項目：
 - 您在 中 AWS 帳戶 建立的叢集清單 AWS ParallelCluster。
 - 所列叢集的可用狀態和詳細資訊。
 - 可用於監控的 CloudFormation 堆疊事件和 AWS ParallelCluster 日誌。
 - 叢集上執行的任務狀態。

- 您可以用來建置叢集的自訂映像清單。
- UI 用來建立叢集的官方映像清單。
- 可存取 PCUI 的使用者清單。您可以新增和移除使用者。
- 提供step-by-step指引。無法存取的輸入欄位無法針對要編輯的叢集組態進行變更。您可以選擇在叢集部署之前執行叢集組態的試執行驗證。
- 具有直接 shell 連結，可在叢集檢視中存取前端節點。在step-by-step指引期間，選擇新增 SSM 工作階段以新增直接 Shell 存取，並在前端節點上選擇 SSM 受管執行個體核心政策。

使用 PCUI 建立和管理叢集時，請考慮下列事項：

- 您只能使用用來建立 PCUI 的相同 AWS ParallelCluster 版本來建立和編輯叢集或建置映像。舊版叢集或映像只能檢視。如果您管理多個版本的叢集和映像，建議您建立 PCUI 執行個體來支援每個版本。
- PCUI 旨在鏡像 CLI `pcluster` 功能。有一些差異。如果您符合step-by-step指引，則會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組態。如果您這樣做，我們建議您選擇「乾執行」來驗證組態，以確認您的編輯完全受支援。

Note

PCUI 不支援 AWS Batch。

AWS Lambda 中的 VPC 組態 AWS ParallelCluster

AWS ParallelCluster 使用 在叢集的生命週期期間 AWS Lambda 執行操作。[AWS Lambda 函數一律會在 Lambda 服務所擁有的 VPC 中執行](#)。此 Lambda 函數也可以連接到虛擬私有雲端 (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。若要連線至專用 VPC 中的資源，請將專用 VPC 與可連線至專用 VPC 的預設租用對等至第二個 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](#)，以及如何[從知識中心將 Lambda 函數連線至專用 VPC ?](#)。Amazon EC2 AWS


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。

- 網際網路存取

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

- VPC 端點

數個 AWS 服務提供 [VPC 端點](#)。您可以使用 VPC 端點 AWS 服務 從沒有網際網路存取的 VPC 連線至。若要檢視 AWS ParallelCluster VPC 端點的清單，請參閱[聯網](#)。

 Note

子網路和安全群組的每個組合都必須 AWS 服務 使用這些方法提供的存取權。子網路和安全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](#)，以及《AWS Lambda 開發人員指南》中 [VPC 連線函數的網際網路和服務存取](#)。

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

AWS Identity and Access Management 中的 許可 AWS ParallelCluster

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

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

- pcluster 使用者叫用 pcluster CLI 命令以建立和管理叢集所需的許可。
- 叢集資源執行叢集動作所需的許可。

AWS ParallelCluster 使用 [Amazon EC2 執行個體描述檔和角色](#) 來提供叢集資源許可。若要管理叢集資源許可，AWS ParallelCluster 也需要 IAM 資源的許可。如需詳細資訊，請參閱 [AWS ParallelCluster 用於管理 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 來建立和管理叢集及其資源。這些許可包含在可新增至使用者或角色的 IAM 政策中。如需 IAM 角色的詳細資訊，請參閱AWS Identity and Access Management 《使用者指南》中的[建立使用者角色](#)。

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

下列各節包含具有範例的必要許可。

若要使用範例政策，請以適當的值取代 `<AWS ACCOUNT ID>`、`<REGION>`和類似的字串。

下列範例政策包含資源的 Amazon Resource Name (ARNs)。如果您在 AWS GovCloud (US) 或 AWS 中國分割區中作業，則必須變更 ARNs。具體而言，它們必須針對 AWS GovCloud (US) 分割區從 "arn:aws" 變更為 "arn:aws-us-gov"，或針對 AWS 中國分割區從 "arn:aws-cn"。如需詳細資訊，請參閱AWS GovCloud (US) 《使用者指南》中的 [AWS GovCloud \(US\) 區域中的 Amazon Resource Name ARNs](#)，以及 [《中國 AWS 服務入門》中的中國服務的 ARNs](#)。AWS

您可以在 [AWS ParallelCluster GitHub 上追蹤文件中範例政策的變更](#)。

主題

- [AWS ParallelCluster Amazon EC2 執行個體角色](#)
- [AWS ParallelCluster 範例pcluster使用者政策](#)
- [AWS ParallelCluster 用於管理 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 執行個體角色

當您使用預設組態設定建立叢集時，AWS ParallelCluster 會使用 Amazon EC2 [執行個體描述](#)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案

您可以使用InstanceRole叢集組態設定來為 EC2 指定自己的現有 IAM 角色，以取代預設 AWS ParallelCluster 執行個體角色。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，以完全控制授予 EC2 的許可。

如果您的意圖是將額外的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時更新，但是，您無法在更新叢集InstanceRole時更新。

AWS ParallelCluster 範例 `pcluster` 使用者政策

下列範例顯示使用 CLI `pcluster` 建立和管理 AWS ParallelCluster 及其資源所需的使用者政策。您可以將政策連接至使用者或角色。

主題

- [基本 AWS ParallelCluster `pcluster` 使用者政策](#)
- [使用 AWS Batch 排程器時的其他 AWS ParallelCluster `pcluster` 使用者政策](#)
- [使用 Amazon FSx for Lustre 時的其他 AWS ParallelCluster `pcluster` 使用者政策](#)
- [AWS ParallelCluster 映像建置 `pcluster` 使用者政策](#)

基本 AWS ParallelCluster `pcluster` 使用者政策

下列政策顯示執行 AWS ParallelCluster `pcluster` 命令所需的許可。

政策中列出的最後一個動作會包含在內，以提供叢集組態中指定之任何秘密的驗證。例如，使用 AWS Secrets Manager 秘密來設定 [DirectoryService](#) 整合。在此情況下，只有在 `SecretArn` 中存在有效的秘密時，才會建立叢集 `PasswordSecretArn`。如果省略此動作，則會略過秘密驗證。為了改善您的安全狀態，建議您僅新增叢集組態中指定的秘密，以縮小此政策陳述式的範圍。

Note

如果現有的 Amazon EFS 檔案系統是叢集中使用的唯一檔案系統，您可以將範例 Amazon EFS 政策陳述式縮小為叢集組態檔案 [SharedStorage](#) 區段的 `efs` 中參考的特定檔案系統。

JSON

```
{
  "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",
        "dynamodb:UntagResource"
    ],
    "Resource": "arn:aws:dynamodb:*:111122223333: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",
        "cloudwatch:TagResource",
        "cloudwatch:UntagResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatch"
},
{
    "Action": [
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetPolicy",
        "iam:SimulatePrincipalPolicy",
        "iam:GetInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::111122223333:role/*",
        "arn:aws:iam::111122223333:policy/*",
        "arn:aws:iam::aws:policy/*",
        "arn:aws:iam::111122223333:instance-profile/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRead"
},
{
    "Action": [
        "iam:CreateInstanceProfile",
        "iam>DeleteInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::111122223333: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:*:111122223333: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:*:111122223333:function:parallelcluster-*",
        "arn:aws:lambda:*:111122223333: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:*:111122223333:*"
      ],
      "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",
        "logs:ListTagsForResource"
      ],
      "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:us-
east-1:111122223333:secret:<SECRET NAME>",
    "Effect": "Allow"
  }
]
}

```

使用 AWS Batch 排程器時的其他 AWS ParallelCluster **pcluster** 使用者政策

如果您需要使用 AWS Batch 排程器建立和管理叢集，則需要以下額外政策。

JSON

```

{
  "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::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": [
          "batch.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:CreateServiceLinkedRole",
      "iam>DeleteServiceLinkedRole"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:role/aws-service-role/
batch.amazonaws.com/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "codebuild:*"
    ],
    "Resource": "arn:aws:codebuild:*:111122223333: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 Amazon FSx for Lustre 時的其他 AWS ParallelCluster **pcluster** 使用者政策

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

Note

如果現有的 Amazon FSx 檔案系統是叢集中使用的唯一檔案系統，您可以將範例 Amazon FSx 政策陳述式縮小為叢集組態檔案 [SharedStorage 區段](#) 的中參考的特定檔案系統。

JSON

```

{
  "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:*:111122223333:*"
  ],
  "Effect": "Allow",
  "Sid": "FSx"
},
{
  "Action": [
    "iam:CreateServiceLinkedRole",
    "iam:AttachRolePolicy",
    "iam:PutRolePolicy"
  ],
  "Resource": "arn:aws:iam::111122223333:role/aws-service-role/s3.data-source.lustre.fsx.amazonaws.com/*",
  "Effect": "Allow"
},
{
  "Action": [
    "s3:Get*",
    "s3:List*",
    "s3:PutObject"
  ],
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
  "Effect": "Allow"
}
```

```
    }  
  ]  
}
```

AWS ParallelCluster 映像建置 `pcluster` 使用者政策

想要使用 建立自訂 Amazon EC2 映像的使用者 AWS ParallelCluster 必須擁有下列一組許可。

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": [  
        "ec2:DescribeSecurityGroups",  
        "ec2:DescribeImages",  
        "ec2:DescribeInstanceTypeOfferings",  
        "ec2:DescribeInstanceTypes",  
        "ec2:DeregisterImage",  
        "ec2:DeleteSnapshot"  
      ],  
      "Resource": "*",  
      "Effect": "Allow",  
      "Sid": "EC2"  
    },  
    {  
      "Action": [  
        "iam:CreateInstanceProfile",  
        "iam:AddRoleToInstanceProfile",  
        "iam:CreateRole",  
        "iam:TagRole",  
        "iam:GetRole",  
        "iam:PutRolePolicy",  
        "iam:GetRolePolicy",  
        "iam:GetInstanceProfile",  
        "iam:RemoveRoleFromInstanceProfile"  
      ],  
      "Resource": [  
        "arn:aws:iam::111122223333:instance-profile/parallelcluster/*",  
        "arn:aws:iam::111122223333:instance-profile/  
ParallelClusterImage*"  
      ]  
    }  
  ]  
}
```

```

        "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IAM"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "lambda.amazonaws.com",
          "ec2.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:instance-profile/parallelcluster/*",
      "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IAMPassRole"
  },
  {
    "Action": [
      "logs:GetLogEvents",
      "logs:CreateLogGroup",
      "logs:TagResource",
      "logs:UntagResource",
      "logs>DeleteLogGroup"
    ],
    "Resource": [
      "arn:aws:logs:*:111122223333:log-group:/aws/imagebuilder/ParallelClusterImage-*",
      "arn:aws:logs:*:111122223333:log-group:/aws/lambda/ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "CloudWatch"
  },
  {
    "Action": [
      "cloudformation:DescribeStacks",

```

```

        "cloudformation:CreateStack",
        "cloudformation>DeleteStack"
    ],
    "Resource": [
        "arn:aws:cloudformation:*:111122223333: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:*:111122223333: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:*:111122223333:image/parallelclusterimage-
*",
        "arn:aws:imagebuilder:*:111122223333:image-recipe/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:111122223333:component/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:111122223333:distribution-configuration/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:111122223333: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:*:111122223333: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 用於管理 IAM 資源的使用者範例政策

使用 AWS ParallelCluster 建立叢集或自訂 AMIs 時，必須提供包含許可的 IAM 政策，以將必要的許可集授予 AWS ParallelCluster 元件。這些 IAM 資源可由自動建立，AWS ParallelCluster 或在建立叢集或自訂映像時提供做為輸入。

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

主題

- [特殊權限 IAM 存取模式](#)
- [受限的 IAM 存取模式](#)
- [PermissionsBoundary 模式](#)

特殊權限 IAM 存取模式

在此模式下，AWS ParallelCluster 會自動建立所有必要的 IAM 資源。這些 IAM 政策的範圍縮小，只能存取叢集資源。

若要啟用特殊權限 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 / Iam SlurmQueues / AdditionalPolicies](#) 參數，您必須為 AWS ParallelCluster 使用者提供為每個額外政策連接和分離角色政策的許可，如下列政策所示。將其他政策 ARNs 新增至連接和分離角色政策的條件。

Warning

此模式可讓使用者在 中擁有 IAM 管理員權限 AWS 帳戶

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:TagRole",
        "iam:UntagRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
      ]
    }
  ]
}
```

```

    "Effect": "Allow",
    "Sid": "IamRole"
  },
  {
    "Action": [
      "iam:CreateRole"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamCreateRole"
  },
  {
    "Action": [
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
  },
  {
    "Condition": {
      "ArnLike": {
        "iam:PolicyARN": [
          "arn:aws:iam::111122223333:policy/parallelcluster*",
          "arn:aws:iam::111122223333: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::111122223333:role/parallelcluster/*",
  "Effect": "Allow",
  "Sid": "IamPolicy"
}
]
}

```

受限的 IAM 存取模式

當沒有將其他 IAM 政策授予使用者時，叢集或自訂映像建置所需的 IAM 角色需要由管理員手動建立，並在叢集組態中傳遞。

建立叢集時，需要下列參數：

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

建置自訂映像時，需要下列參數：

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

作為上述所列參數的一部分傳遞的 IAM 角色必須在/parallelcluster/路徑字首上建立。如果無法這麼做，則需要更新使用者政策，才能授予特定自訂角色的iam:PassRole許可，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 角色都可以在叢集組態中傳遞。

PermissionsBoundary 模式

此模式會委派 AWS ParallelCluster 建立繫結至已設定 IAM 許可界限的 IAM 角色。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](#)。

下列政策需要新增至使用者角色。

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 / Iam / AdditionalPolicies](#) 或 [Scheduling / Iam SlurmQueues / / AdditionalPolicies](#) 參數，您必須授予使用者許可，以連接和分離每個額外政策的角色政策，如下列政策所示。將其他政策 ARNs 新增至連接和分離角色政策的條件。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:TagRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRole"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PermissionsBoundary": [
            "<permissions-boundary-arn>"
          ]
        }
      },
      "Action": [
        "iam>CreateRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamCreateRole"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PermissionsBoundary": [
            "<permissions-boundary-arn>"
          ]
        }
      },
      "Action": [
```

```

        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
},
{
    "Condition": {
        "StringEquals": {
            "iam:PermissionsBoundary": [
                "<permissions-boundary-arn>"
            ]
        },
        "ArnLike": {
            "iam:PolicyARN": [
                "arn:aws:iam::111122223333:policy/parallelcluster*",
                "arn:aws:iam::111122223333: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::111122223333: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](#) 組態參數中指定許可界限 ARN，並在建置自訂映像時在 [Build / Iam / PermissionBoundary](#) 參數中指定許可界限 ARN。

AWS ParallelCluster 用於管理 IAM 許可的組態參數

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

主題

- [叢集組態](#)
- [自訂映像組態](#)

叢集組態

主題

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

前端節點 IAM 角色

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

使用此選項，您可以覆寫指派給叢集前端節點的預設 IAM 角色。如需其他詳細資訊，請參閱 [InstanceProfile](#) 參考。

以下是排程器為 Slurm 時，要做為此角色一部分使用的一組最少政策：

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::us-east-1-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.us-east-1/*",
        "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:us-east-1:111122223333: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::111122223333:role/parallelcluster/*",
      "arn:aws:iam::111122223333: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:us-east-1:111122223333:instance/*",
      "arn:aws:ec2:us-east-1:111122223333: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:us-
east-1:111122223333:secret:<SECRET_ID>",
    "Effect": "Allow"
  }
]
}

```

請注意，如果 [Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) 用於覆寫運算 IAM 角色，上述報告的前端節點政策需要在 iam:PassRole 許可的 Resource 區段中包含此類角色。

以下是排程器為 時，要做為此角色一部分使用的一組最少政策 AWS Batch：

- arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy 受管 IAM 政策。如需詳細資訊，請參閱《Amazon [CloudWatch 使用者指南](#)》中的 [建立 IAM 角色和使用者以搭配 CloudWatch 代理程式使用](#)。Amazon CloudWatch

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 受管 IAM 政策。如需詳細資訊，請參閱AWS Systems Manager 《使用者指南》中的 [AWS 的受管政策 AWS Systems Manager](#)。
- 其他 IAM 政策：

JSON

```
{
  "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::dcl-license.us-east-1/*",
        "arn:aws:s3:::us-east-1-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/parallelcluster/*",
        "arn:aws:iam::111122223333: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:us-east-1:111122223333:log-group:/aws/batch/
      job:log-stream:PclusterJobDefinition*",
      "arn:aws:ecs:us-east-1:111122223333:container-instance/
      AWSBatch-PclusterComputeEnviron*",
      "arn:aws:ecs:us-east-1:111122223333:cluster/AWSBatch-
      Pcluster*",
      "arn:aws:batch:us-east-1:111122223333:job-queue/
      PclusterJobQueue*",
      "arn:aws:batch:us-east-1:111122223333:job-definition/
      PclusterJobDefinition*:*",
      "arn:aws:batch:us-east-1:111122223333:job/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeVolumes",
      "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*"
  }

```

```

    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:CreateTags",
      "ec2:AttachVolume"
    ],
    "Resource": [
      "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "arn:aws:ec2:us-east-1:111122223333:volume/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStacks",
      "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:<SECRET_ID>",
    "Effect": "Allow"
  }
]
}

```

Amazon S3 存取

[HeadNode](#) / [Iam](#) / [S3Access](#) 或 [Scheduling](#) / [SlurmQueues](#) / [S3Access](#)

在這些組態區段中，當建立 Amazon S3 角色時，您可以將其他 Amazon S3 政策授予與叢集前端節點或運算節點相關聯的 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。

只有在使用 [特殊權限 IAM 存取模式](#) 或設定使用者時，才能使用此參數 [PermissionsBoundary 模式](#)。

其他 IAM 政策

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

使用此選項，將其他受管 IAM 政策連接至建立此類角色時與叢集前端節點或運算節點相關聯的 IAM 角色 AWS ParallelCluster。

Warning

若要使用此選項，請確定已授予 [AWS ParallelCluster 使用者](#) `iam:AttachRolePolicy` 和需要連接之 IAM 政策的 `iam:DetachRolePolicy` 許可。

AWS Lambda 函數角色

[Iam / Roles / LambdaFunctionsRole](#)

此選項會覆寫連接到叢集建立程序期間使用之所有 AWS Lambda 函數的角色。AWS Lambda 需要設定為允許擔任該角色的委託人。

Note

如果 [LambdaFunctionsVpcConfig](#) 設定 [DeploymentSettings /](#)，[LambdaFunctionsRole](#) 必須包含 [AWS Lambda 角色許可](#)，才能設定 VPC 組態。

以下是要做為此角色一部分使用的一組最少政策：

JSON

```
{
  "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:us-east-1:111122223333: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](#)

此選項可讓 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](#)。

以下是要做為此角色一部分使用的一組最少政策：

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

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:Query",
        "dynamodb:UpdateItem",
        "dynamodb:PutItem",
        "dynamodb:GetItem"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:111122223333:table/parallelcluster-*",
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3::us-east-1-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "ec2:DescribeInstanceAttribute",
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": "cloudformation:DescribeStackResource",
      "Resource": [
```

```
        "arn:aws:cloudformation:us-east-1:111122223333:stack/*/*"
    ],
    "Effect": "Allow"
  }
]
}
```

許可界限

[Iam / PermissionsBoundary](#)

此參數 AWS ParallelCluster 會強制將指定的 IAM 政策做為 PermissionsBoundary 連接到建立為叢集部署一部分的所有 IAM 角色。

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

自訂映像組態

主題

- [EC2 Image Builder 的執行個體角色](#)
- [AWS Lambda 清除角色](#)
- [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 Amazon EC2 EC2 執行個體的 IAM 角色，以建立自訂 AMI。

以下是要做為此角色一部分使用的一組最少政策：

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 受管 IAM 政策。如需詳細資訊，請參閱AWS Systems Manager 《使用者指南》中的 [AWS 的受管政策 AWS Systems Manager](#)。
- `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder` 受管 IAM 政策。如需詳細資訊，請參閱《映像建置器使用者指南》中的[EC2InstanceProfileForImageBuilder政策](#)。
- 其他 IAM 政策：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateTags",
        "ec2:ModifyImageAttribute"
      ],
      "Resource": "arn:aws:ec2:us-east-1:image/*",
      "Effect": "Allow"
    }
  ]
}
```

AWS Lambda 清除角色

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

此選項會覆寫附加至自訂映像建置程序期間使用之所有 AWS Lambda 函數的角色。AWS Lambda 需要設定為允許擔任該角色的主體。

Note

如果 [LambdaFunctionsVpcConfig](#) 設定 [DeploymentSettings](#) / ,
CleanupLambdaRole 必須包含 [AWS Lambda 角色許可](#) , 才能設定 VPC 組態。

以下是要做為此角色一部分使用的一組最少政策：

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

JSON

```
{
```



```

"Version": "2012-10-17",
"Statement": [
  {
    "Action": [
      "iam:DetachRolePolicy",
      "iam>DeleteRole",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::111122223333:role/parallelcluster/*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam>DeleteInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": "arn:aws:iam::111122223333:instance-profile/parallelcluster/*",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder>DeleteInfrastructureConfiguration",
    "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:infrastructure-configuration/parallelclusterimage-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder>DeleteComponent"
    ],
    "Resource": [
      "arn:aws:imagebuilder:us-east-1:111122223333:component/parallelclusterimage-*/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder>DeleteImageRecipe",
    "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:image-recipe/parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder>DeleteDistributionConfiguration",

```

```

        "Resource": "arn:aws:imagebuilder:us-
east-1:111122223333:distribution-configuration/parallelclusterimage-*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "imagebuilder:DeleteImage",
            "imagebuilder:GetImage",
            "imagebuilder:CancelImageCreation"
        ],
        "Resource": "arn:aws:imagebuilder:us-east-1:111122223333:image/
parallelclusterimage-*/**",
        "Effect": "Allow"
    },
    {
        "Action": "cloudformation:DeleteStack",
        "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*/*",
        "Effect": "Allow"
    },
    {
        "Action": "ec2:CreateTags",
        "Resource": "arn:aws:ec2:us-east-1::image/*",
        "Effect": "Allow"
    },
    {
        "Action": "tag:TagResources",
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "lambda:DeleteFunction",
            "lambda:RemovePermission"
        ],
        "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:ParallelClusterImage-*",
        "Effect": "Allow"
    },
    {
        "Action": "logs:DeleteLogGroup",
        "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
lambda/ParallelClusterImage-*:*",
        "Effect": "Allow"
    }

```

```
    },
    {
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:DeleteTopic",
        "SNS:GetSubscriptionAttributes",
        "SNS:Unsubscribe"
      ],
      "Resource": "arn:aws:sns:us-  
east-1:111122223333: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 政策連接至與 Amazon EC2 EC2 執行個體相關聯的角色。

Warning

若要使用此選項，請確定已授予[AWS ParallelCluster使用者](#)iam:AttachRolePolicy和需要連接之 IAM 政策的iam:DetachRolePolicy許可。

許可界限

[Build / Iam / PermissionsBoundary](#)

此參數 AWS ParallelCluster 會強制將指定的 IAM 政策做為 PermissionsBoundary 連接到建立為自訂 AMI 組建一部分的所有 IAM 角色。

如需使用此類功能所需的政策清單，[PermissionsBoundary 模式](#)請參閱。

網路組態

AWS ParallelCluster 使用 Amazon Virtual Private Cloud (VPC) 進行聯網。VPC 提供靈活且可設定的聯網平台，您可以在其中部署叢集。

VPC 必須具有 DNS Resolution = yes、DNS Hostnames = yes 和 DHCP 選項，且該選項具有區域的正確網域名稱。預設 DHCP 選項集已指定必要的 AmazonProvidedDNS。如果指定多個網域名稱伺服器，請參閱《Amazon VPC 使用者指南》中的 [DHCP 選項集](#)。

AWS ParallelCluster 支援下列高階組態：

- 一個用於頭部和運算節點的子網路。
- 兩個子網路，一個公有子網路中的前端節點，以及私有子網路中的運算節點。子網路可以是新的或現有的子網路。

所有這些組態都可以搭配或不搭配公有 IP 定址運作。AWS ParallelCluster 也可以部署為針對所有 AWS 請求使用 HTTP 代理。這些組態的組合會產生許多部署案例。例如，您可以設定透過網際網路進行所有存取的單一公有子網路。或者，您可以使用 AWS Direct Connect 和 HTTP 代理為所有流量設定全私有網路。

從 AWS ParallelCluster 3.0.0 開始，您可以為每個佇列設定不同的 SecurityGroupsAdditionalSecurityGroups 和 PlacementGroup 設定。如需詳細資訊，請參閱 [HeadNode / Networking](#) 和 [SlurmQueues / Networking](#) 和 [AwsBatchQueues / Networking](#)。

如需某些聯網案例的圖例，請參閱下列架構圖。

主題

- [AWS ParallelCluster 在單一公有子網路中](#)
- [AWS ParallelCluster 使用兩個子網路](#)
- [AWS ParallelCluster 在單一私有子網路中，使用連線 AWS Direct Connect](#)
- [AWS ParallelCluster 使用 AWS Batch 排程器](#)
- [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#)

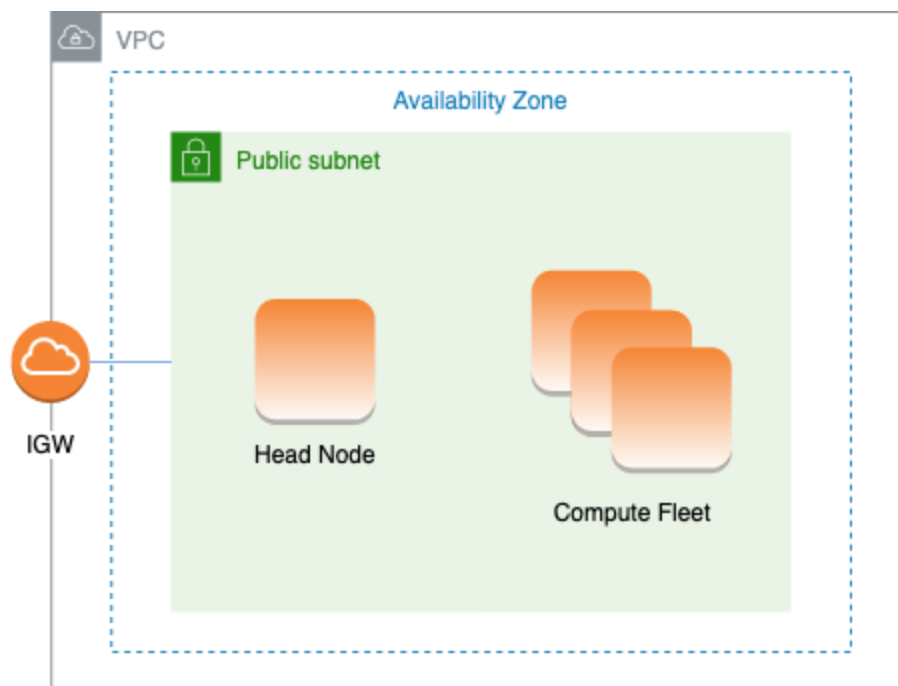
AWS ParallelCluster 在單一公有子網路中

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

- 為用於 // 的子網路開啟「啟用自動指派公有 IPv4 地址」設定，或在 [HeadNode / HeadNode](#) / 中指派彈性 IP [NetworkingSubnetId](#)，以確保前端節點已指派公有 IP [Networking](#) 地址 [ElasticIp](#)。

- 為 `///` 中使用的子網路開啟「啟用自動指派公有 IPv4 地址」設定，或在 `/Scheduling/SlurmQueues/Networking` 中 `SubnetIds` 設定 `AssignPublicIp: true`，以確保將公有 IP `Scheduling` 地址指派給運算節點 `SlurmQueuesNetworking`。
- 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到前端節點，您必須將 `HeadNode/Networking/ElasticIp` 設定為 `true` 以提供公有存取。AWS 公有 IPs 只能指派給使用單一網路介面啟動的執行個體。在此情況下，我們建議您使用 [NAT 閘道](#) 來提供叢集運算節點的公有存取權。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》中的在執行個體 [啟動期間指派公有 IPv4](#) 地址。Amazon EC2
- 您無法定義 `p4d` 或 `hp6id` 執行個體類型，或具有多個網路介面或網路介面卡來運算節點的另一個執行個體類型，因為 AWS 公有 IPs 只能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》中的在執行個體 [啟動期間指派公有 IPv4](#) 地址。Amazon EC2

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](#)。



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
```

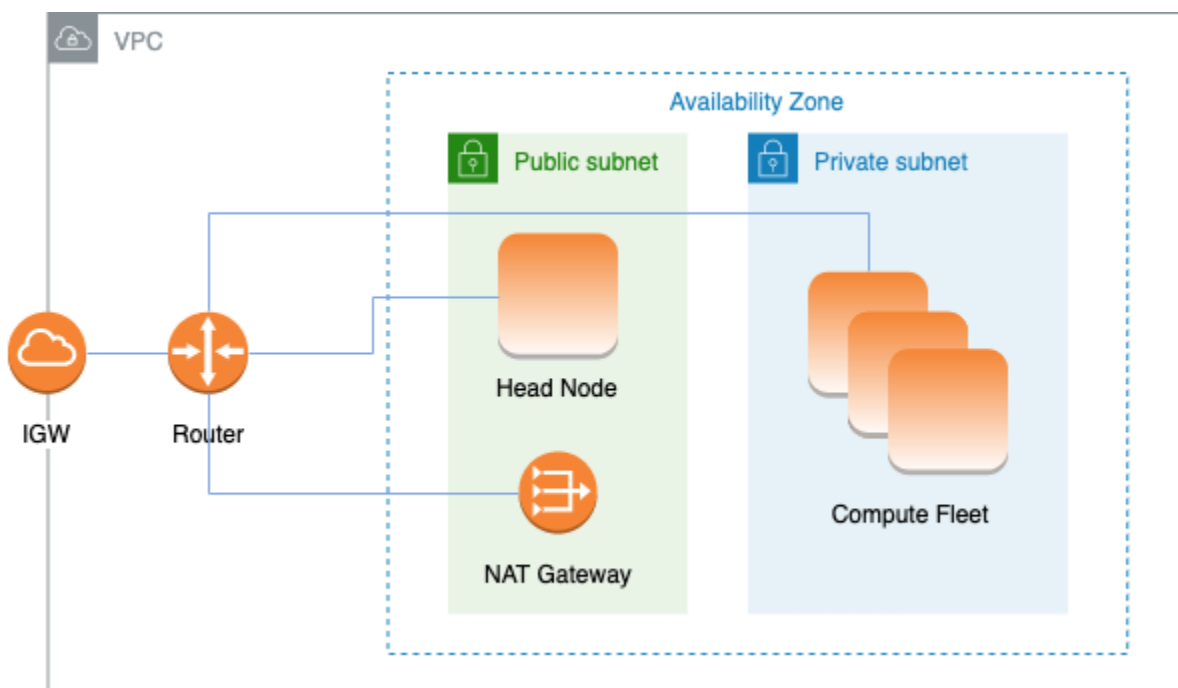
```
#ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
  Networking:
    SubnetIds:
      - subnet-12345678 # subnet with internet gateway
    #AssignPublicIp: true
```

AWS ParallelCluster 使用兩個子網路

在此組態中，只有叢集的前端節點需要指派公有 IP。您可以為 // [HeadNode](#) 中使用的子網路開啟「啟用自動指派公有 IPv4 [Networking](#) 地址」設定，[SubnetId](#)或在 / [HeadNode](#) / [Networking](#) 中指派彈性 IP 來達成此目的[ElasticIp](#)。

如果您定義具有多個網路介面或網路介面卡的 p4d 執行個體類型或其他執行個體類型到前端節點，您必須將 [HeadNode](#) / [Networking](#) / [ElasticIp](#) 設定為 true 以提供公有存取。AWS 公有 IPs 只能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》中的在執行個體[啟動期間指派公有 IPv4](#) 地址。Amazon EC2

此組態需要 [NAT 閘道](#) 或用於佇列之子網路中的內部代理，才能讓網際網路存取運算執行個體。

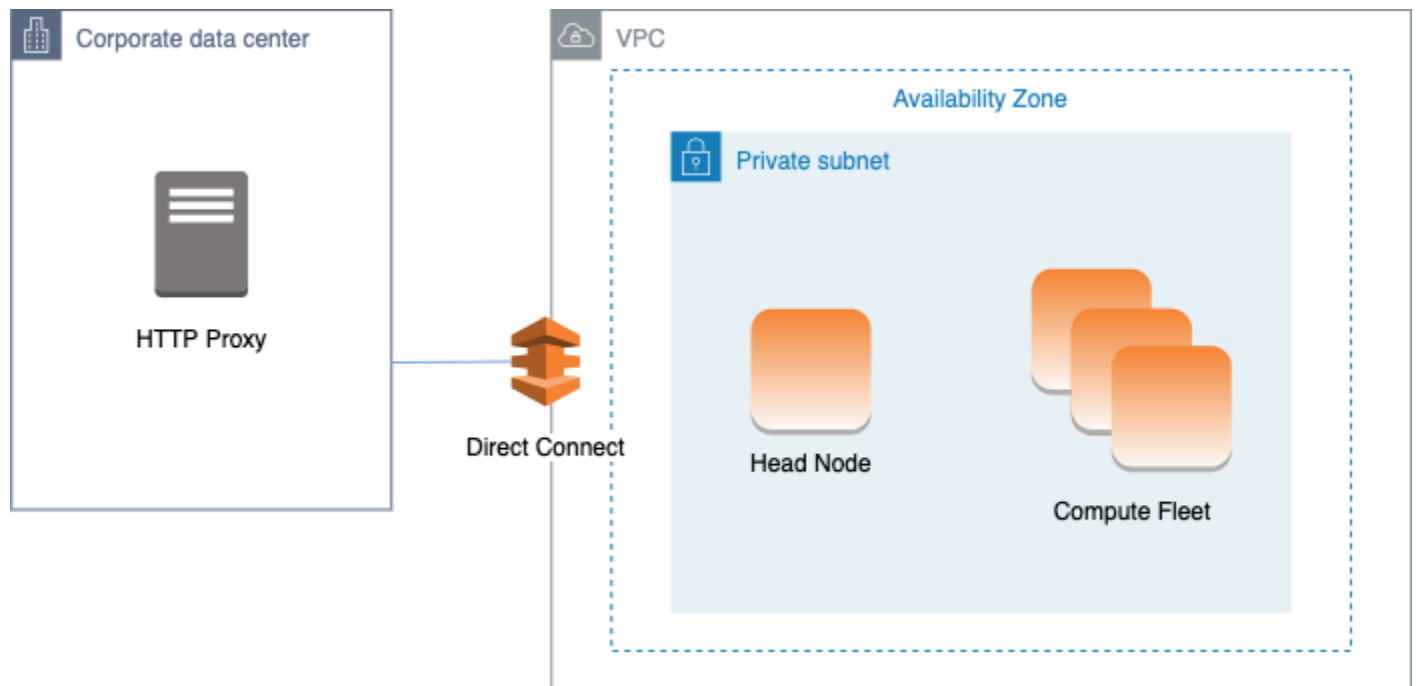


針對運算執行個體使用現有私有子網路的組態需要下列設定：

```
# 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
```

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

當 [Scheduling](#) / [SlurmQueues](#) / [Networking](#) / [AssignPublicIp](#) 設定為 `false`，子網路必須正確設定，才能對所有流量使用 Proxy。頭部和運算節點都需要 Web 存取。



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
```

AWS ParallelCluster 使用 AWS Batch 排程器

當您使用 `awsbatch` 做為排程器類型時，會 AWS ParallelCluster 建立 AWS Batch 受管運算環境。AWS Batch 環境會管理 Amazon Elastic Container Service (Amazon ECS) 容器執行個體。這些執行個體會在 / [AwsBatchQueues Networking/ SubnetIds](#) 參數中設定的子網路中啟動。為了 AWS Batch 讓正常運作，Amazon ECS 容器執行個體需要外部網路存取，才能與 Amazon ECS 服務端點通訊。這轉換成以下案例：

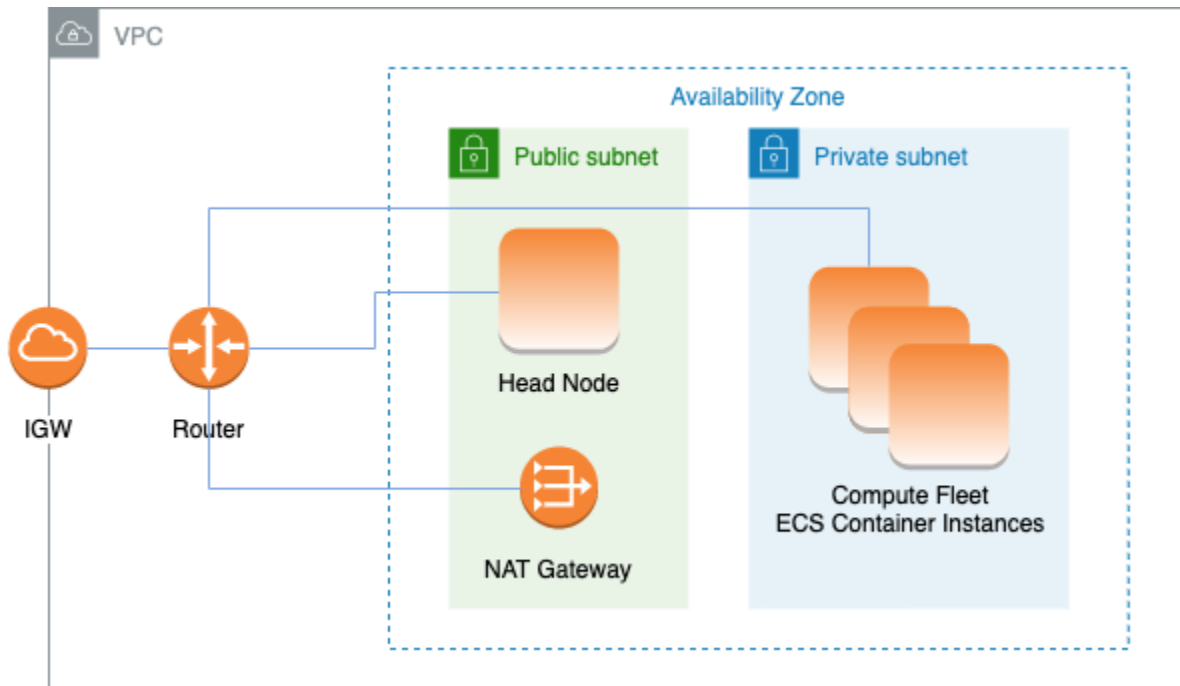
- 為佇列指定的子網路 ID 會使用 [NAT 閘道](#) 來存取網際網路。我們建議使用此方法。
- 在佇列子網路中啟動的執行個體具有公有 IP 地址，並且可以透過網際網路閘道連線到網際網路。

此外，如果您對多節點平行任務感興趣（來自 [AWS Batch 文件](#)）：

AWS Batch 多節點平行任務使用 Amazon ECS `awsvpc` 網路模式。這為您的多節點平行任務容器提供與 Amazon EC2 執行個體相同的聯網屬性。每個多節點平行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。網路界面是在與其主機運算資源相同的 Amazon VPC 子網路中建立。任何套用到您運算資源的安全群組，也會套用在它身上。

使用 Amazon ECS 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 地址的彈性網路介面。若要存取網際網路，必須在設定為使用 NAT 閘道的私有子網路中啟動使用 Amazon EC2 啟動類型的任務。

您必須設定 [NAT 閘道](#)，才能讓叢集執行多節點平行任務。



所有先前的組態和考量 AWS Batch 也適用於。以下是 AWS Batch 聯網組態的範例。

```
# 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，但目前僅支援一個子網路。

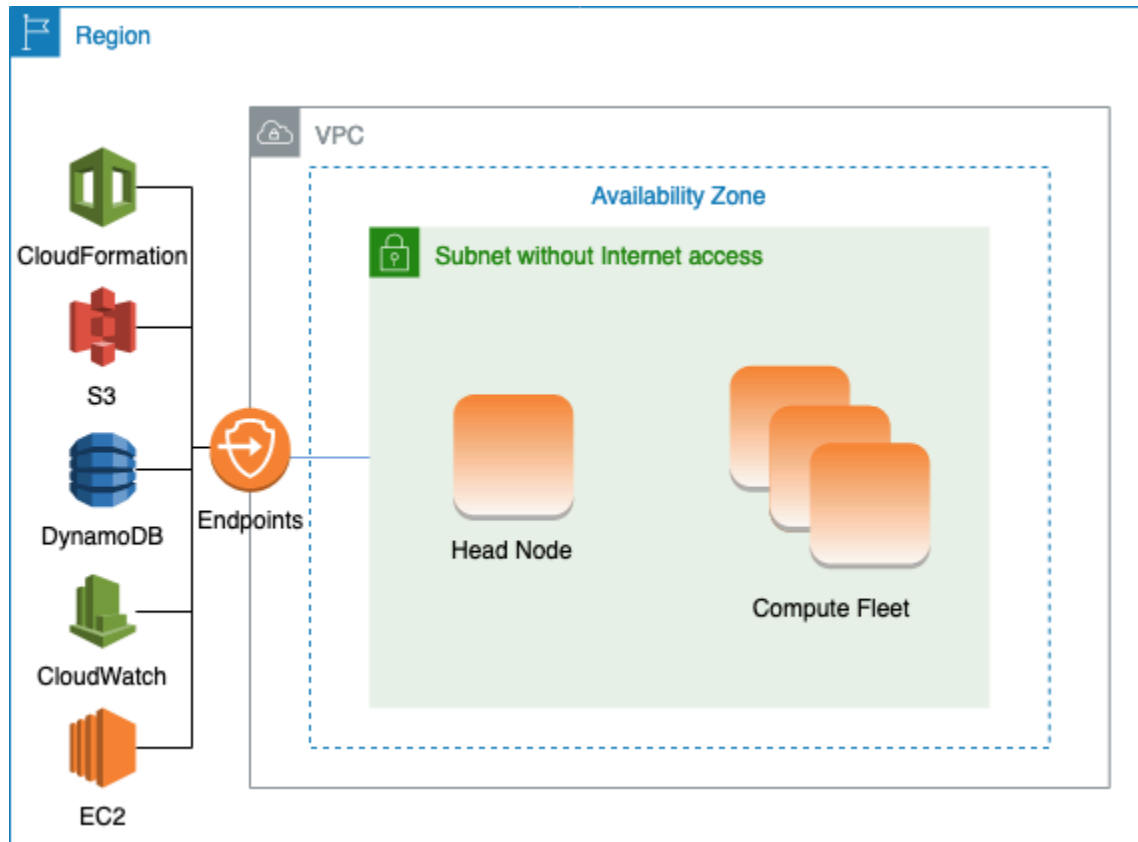
如需詳細資訊，請參閱下列主題：

- [AWS Batch 受管運算環境](#)
- [AWS Batch 多節點平行任務](#)
- [使用 awsipc 網路模式的 Amazon ECS 任務聯網](#)

AWS ParallelCluster 在沒有網際網路存取的單一子網路中

沒有網際網路存取的子網路不允許網際網路的傳入或傳出連線。此 AWS ParallelCluster 組態可協助擔心安全性的客戶進一步增強其 AWS ParallelCluster 資源的安全性。AWS ParallelCluster 節點是從 AWS ParallelCluster AMIs 建置的，其中包含在無網際網路存取的情況下執行叢集所需的所有軟體。如此一來，AWS ParallelCluster 就可以建立和管理具有無法存取網際網路之節點的叢集。

在本節中，您將了解如何設定叢集。您也了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

為了確保叢集正常運作，叢集節點必須能夠與多個 AWS 服務互動。

建立和設定下列 [VPC 端點](#)，讓叢集節點可以與 AWS 服務互動，無需網際網路存取：

Commercial and AWS GovCloud (US) partitions

服務	服務名稱	Type
Amazon CloudWatch	com.amazonaws. <i>region-id</i> .logs	介面
CloudFormation	com.amazonaws. <i>region-id</i> .cloudformation	介面
Amazon EC2	com.amazonaws. <i>region-id</i> .ec2	介面
Amazon S3	com.amazonaws. <i>region-id</i> .s3	閘道
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	閘道
AWS Secrets Manager**	com.amazonaws. <i>region-id</i> .secretsmanager	介面
AWS Elastic Load Balancing***	com.amazonaws. <i>region-id</i> .elasticloadbalancing	介面
AWS Auto Scaling***	com.amazonaws. <i>region-id</i> .autoscaling	介面

China partition

服務	服務名稱	Type
Amazon CloudWatch	com.amazonaws. <i>region-id</i> .logs	介面
CloudFormation	cn.com.amazonaws. <i>region-id</i> .cloudformation	介面

服務	服務名稱	Type
Amazon EC2	cn.com.am azonaws. <i>region-id</i> .ec2	介面
Amazon S3	com.amazonaws. <i>region-id</i> .s3	閘道
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	閘道
AWS Secrets Manager**	com.amazonaws. <i>region-id</i> .secretsmanager	介面
AWS Elastic Load Balancing ***	com.amazonaws. <i>region-id</i> .elasticloadbalancing	介面
AWS Auto Scaling***	cn.com.am azonaws. <i>region-id</i> .autoscaling	介面

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

*** 只有在已啟用 [LoginNodes](#) 時，才需要這些端點，否則它們是選用的。

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。您可以透過將安全群組新增至 [AdditionalSecurityGroups](#) 下的 [HeadNode](#) 和 [SlurmQueues](#) 組態 [AdditionalSecurityGroups](#) 下的 來執行此操作。例如，如果建立 VPC 端點時未明確指定安全群組，則預設安全群組會與端點建立關聯。透過將預設安全群組新增至 [AdditionalSecurityGroups](#)，您可以啟用叢集與端點之間的通訊。

Note

當您使用 IAM 政策來限制對 VPC 端點的存取時，您必須將下列項目新增至 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 Route 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，用於解析自訂運算節點主機名稱，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。將下列設定套用至您的叢集組態：

```

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

```

Warning

對於使用 [SlurmSettings // Dns](#) 建立 [DisableManagedDns](#) 並 [UseEc2Hostnames](#) 設為 `true` 的叢集，DNS Slurm NodeName 不會解析。請 `SlurmNodeHostName` 改用。

Note

此備註從 3.3.0 AWS ParallelCluster 版開始並不相關。

對於 3.3.0 之前的 AWS ParallelCluster 支援版本：

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

- `prolog` 會執行 以在每個任務配置時，將節點資訊新增至運算節點/`etc/hosts`上的。
- `epilog` 會執行 以清除 撰寫的內容 `prolog`。

若要新增自訂prolog或epilog指令碼，請分別將它們新增至 /opt/slurm/etc/pcluster/prolog.d/或 /opt/slurm/etc/pcluster/epilog.d/ 資料夾。

叢集組態

了解如何將叢集設定為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
LoginNodes: # optional, if enabled, requires creation and configuration of VPC
endpoints for AWS Elastic Load Balancing (ELB) and Auto Scaling services
Pools:
  - ...
    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
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\(s\)](#) : 沒有網際網路存取之子網路。

若要啟用 AWS ParallelCluster 和 AWS 服務之間的通訊，子網路的 VPC 必須連接 VPC 端點。建立叢集之前，請確認子網路中 [已停用自動指派公有 IPv4 地址](#)，以確保 pcluster 命令可存取叢集。

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

選用：

- 如果建立 VPC 端點時未明確指定安全群組，則會關聯 VPC 的預設安全群組。因此，請將預設安全群組提供給 AdditionalSecurityGroups。
- 如果在建立叢集和/或 VPC 端點時使用自訂安全群組，只要自訂安全群組啟用叢集和 VPC 端點之間的通訊，AdditionalSecurityGroups 則不需要。

- [Scheduler](#) : 叢集排程器。

slurm 是唯一的有效值。只有 Slurm 排程器支援子網路中沒有網際網路存取權的叢集。

- [SlurmSettings](#) : Slurm 設定。

請參閱上一節停用 Route53 並使用 Amazon EC2 主機名稱。

限制

- 透過 SSH 或 Amazon DCV 連線至前端節點：連線至叢集時，請確定連線的用戶端可以透過其私有 IP 地址連線到叢集的前端節點。如果用戶端不在與前端節點相同的 VPC 中，請在 VPC 的公有子網路中使用代理執行個體。此要求同時適用於 SSH 和 DCV 連線。如果子網路沒有網際網路存取，則無法存取前端節點的公有 IP。如果公有 IP 或私有 IP，pcluster ssh 和 dcv-connect 命令會使用該公有 IP。建立叢集之前，請確認子網路中 [已停用自動指派公有 IPv4 地址](#)，以確保 pcluster 命令可存取叢集。

下列範例示範如何連線到叢集前端節點中執行的 DCV 工作階段。您可以透過代理 Amazon EC2 執行個體進行連線。執行個體可做為 PC 的 Amazon DCV 伺服器，以及私有子網路中前端節點的用戶端。

透過公有子網路中的代理執行個體透過 DCV 連線：

1. 在與叢集子網路位於相同 VPC 的公有子網路中建立 Amazon EC2 執行個體。
 2. 確定 Amazon 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 使用者政策連接至代理 Amazon EC2 執行個體。如需詳細資訊，請參閱[AWS ParallelCluster 範例 pcluster 使用者政策](#)。
 4. 在代理 Amazon EC2 執行個體 AWS ParallelCluster 上安裝。
 5. 透過 DCV 連線至代理 Amazon EC2 執行個體。
 6. 在代理執行個體上使用 `pcluster dcv-connect` 命令，在沒有網際網路存取的情況下連線到子網路內的叢集。
- 與其他 AWS 服務互動：上面僅 AWS ParallelCluster 列出 嚴格要求的服務。如果您的叢集必須與其他服務互動，請建立對應的 VPC 端點。

佈建的登入節點 AWS ParallelCluster

從 3.7.0 版開始，AWS ParallelCluster 叢集管理員可以佈建登入節點，用來提供使用者執行任務的存取權，而不是直接存取叢集前端節點。具有適當許可的叢集使用者可以使用 Active Directory 或其 SSH 登入資料來登入、提交和管理其任務。因此，可以改善叢集管理，而且可以將 Slurm 管理叢集所需的前端節點資源消耗的機會降至最低。登入的使用者也可以存取掛載在登入節點上的叢集的所有共用儲存。如果需要停止登入節點，登入的使用者將透過其正在使用的作用中 shell 工作階段預先收到通知。

登入節點指定為集區，其中集區會定義具有相同資源組態的登入節點群組。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，以循環方式跨登入節點分配工作階段。目前的實作允許使用者指定多個登入節點集區。

登入節點的安全性

登入節點 [AllowedIps](#) 會從前端節點繼承 AllowedIPs 設定，除非為 [登入節點集區](#) 指定 AllowedIps。以這種方式，叢集管理員可以透過指定來源 CIDR 或字首清單來限制叢集的安全狀態，在前端節點或登入節點集區上允許 SSH 連線。

目前實作中，啟用登入節點時不會自動限制對前端節點的存取。如有需要，叢集管理員可以使用標準 Linux 命令來限制此存取更新前端節點 SSH 組態。您也可以使用 ParallelCluster YAML 檔案的前端節點區段中的 `AdditionalSecurityGroups` 設定，拒絕來自未經授權使用者的連線，在前端節點上指定自訂安全群組來完成此操作。

登入節點的網路

登入節點是以單一連線地址佈建到為登入節點集區設定的網路負載平衡器。地址的連線設定是根據登入節點集區組態中指定的子網路類型。

- 如果子網路是私有的，地址將是私有的，而且為了授予登入節點的存取權，叢集管理員必須佈建堡壘主機。
- 如果子網路為公有，則該地址將為公有

所有連線請求都是由 Network Load Balancer 使用循環路由來管理。

登入節點的儲存體

在叢集上使用 ParallelCluster 設定的所有共用儲存，包括受管儲存，都會掛載在所有登入節點上。

擷取登入節點資訊

若要擷取佈建以存取登入節點的單一連線地址，叢集管理員可以執行 [describe-cluster](#) 命令。命令也會提供有關登入節點狀態的詳細資訊。

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

登入節點集區的單一連線地址可用性不會阻止直接存取特定的登入節點。不過，不建議使用直接連線，以避免來自 SSH 用戶端的警告。SSH 用戶端會將每個目標地址的主機識別符存放在本機。由於主機識別符是特定於每個集區，因此使用不同的 IPs 和/或單一連線地址可能會有與不同目標地址相關聯的相同主機識別符：這可能會導致 SSH 用戶端警告，因為相同的主機識別符與多個目標相關聯。

登入節點的 Imds 屬性

對登入節點 IMDS（和執行個體設定檔登入資料）的存取僅限於 根使用者、叢集管理使用者（`pc-cluster-admin` 預設）和作業系統特定的預設使用者（在 Amazon Linux 2 和 RedHat `ec2-user` 上，以及在 Ubuntu 18.04 `ubuntu` 上）。

若要限制 IMDS 存取，AWS ParallelCluster 會管理 鏈 `iptables`。

Note

任何自訂 `iptables` 或 `ip6tables` 規則都會干擾用於限制登入節點 IMDS 存取的機制。另請參閱 [Imds property setting](#)。

登入節點生命週期

目前，沒有專用命令可停止和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指定登入節點計數為零的叢集組態Count: 0 ()，然後執行 [pcluster.update-cluster-v3](#) 命令。

Note

登入的使用者會收到有關特定執行個體終止和相關寬限期的通知。在寬限期內，除了來自叢集預設使用者的連線之外，不允許新的連線。顯示的訊息可由叢集管理員從前端節點或從編輯檔案的登入節點自訂/opt/parallelcluster/shared_login_nodes/loginmgtd_config.json。當您使用 [AWS Systems Manager Session Manager](#) Session Manager 連線時，不會顯示此終止訊息。

若要啟動登入節點集區，叢集管理員必須還原叢集組態中的先前Count 值，然後執行 [update-cluster](#) 命令。

執行登入節點集區所需的許可

為了管理登入節點集區，叢集管理員必須具有下列其他許可：

```
- Action:
  - iam:CreateServiceLinkedRole
  - 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
    - autoscaling>DeleteTags
    - autoscaling>CreateOrUpdateTags
    - elasticloadbalancing:AddTags
    - elasticloadbalancing>CreateLoadBalancer
    - elasticloadbalancing:RemoveTags
    - elasticloadbalancing:ModifyTargetGroup
  Resource: '*'
  Effect: Allow
```

自訂引導操作

如果您定義 [HeadNode](#) / [CustomActions](#) / [OnNodeStart](#) 組態設定，會在節點啟動後立即 AWS ParallelCluster 執行任意程式碼。如果您定義 [HeadNode](#) / [CustomActions](#) / [OnNodeConfigured](#) 組態設定，會在節點組態正確完成後 AWS ParallelCluster 執行程式碼。

從 3.4.0 AWS ParallelCluster 版開始，如果您定義 [CustomActions](#) / [HeadNode](#) / [OnNodeUpdated](#) 組態設定，則可以在前端節點更新後執行程式碼。

在大多數情況下，此程式碼會儲存在 Amazon Simple Storage Service (Amazon S3) 中，並透過 HTTPS 連線存取。程式碼會以執行 root，並且可以使用叢集作業系統支援的任何指令碼語言。程式碼通常使用 Bash 或 Python。

Note

從 3.7.0 [Imds](#) AWS ParallelCluster 版開始，叢集/[ImdsSupport](#) 預設設定為 v2.0。當您建立新的叢集以升級至 3.7.0 版和更新版本時，請更新您的自訂引導操作指令碼，使其與 IMDSv2 相容，或在 v1.0 叢集組態檔案中將 [Imds](#) / [ImdsSupport](#) 設定為。

Warning

您有責任設定自訂指令碼和引數，如 [共同責任模型](#) 中所述。確認您的自訂引導指令碼和引數來自您信任的來源，以完整存取叢集節點。

⚠ Warning

AWS ParallelCluster 不支援使用透過 `/etc/parallelcluster/cfnconfig` 檔案提供的內部變數。此檔案可能在未來版本中移除。

`OnNodeStart` 動作會在啟動任何節點部署引導操作之前呼叫，例如設定 NAT、Amazon Elastic Block Store (Amazon EBS) 或排程器。`OnNodeStart` 引導操作可能包括修改儲存、新增額外的使用者，以及新增套件。

📘 Note

如果您為叢集設定 [HeadNode DirectoryService](#) 和 // [CustomActions OnNodeStart](#) 指令碼 `sssd`，會在執行 `OnNodeStart` 指令碼之前 AWS ParallelCluster 設定 `DirectoryService` 並重新啟動。

`OnNodeConfigured` 動作會在節點引導程序完成後呼叫。`OnNodeConfigured` 動作提供執行個體視為完全設定並完成之前要執行的最後一個動作。有些 `OnNodeConfigured` 動作包括變更排程器設定、修改儲存體和修改套件。您可以在組態期間指定引數，將引數傳遞至指令碼。

`OnNodeUpdated` 動作會在前端節點更新完成後呼叫，且排程器和共用儲存與最新的叢集組態變更保持一致。

當 `OnNodeStart` 或 `OnNodeConfigured` 自訂動作成功時，成功會以結束碼零 (0) 表示。任何其他結束碼表示執行個體引導失敗。

當 `OnNodeUpdated` 自訂動作成功時，成功會以結束碼零 (0) 發出訊號。任何其他結束碼表示更新失敗。

📘 Note

如果您設定 [OnNodeUpdated](#)，則必須在更新失敗時手動將 `OnNodeUpdated` 動作還原至先前的狀態。

如果 `OnNodeUpdated` 自訂動作失敗，更新會回復到先前的狀態。不過，`OnNodeUpdated` 動作只會在更新時間執行，而不會在堆疊復原時間執行。

您可以在 `/` 和 `i` [HeadNode](#) // [CustomActionsSchedulingSlurmQueuesCustomActions](#) 組態區段中為每個佇列指定不同的指令碼。 [OnNodeUpdated](#) 只能在 HeadNode 區段中設定。

Note

在 3.0 AWS ParallelCluster 版之前，無法為頭部和運算節點指定不同的指令碼。請參閱 [從 AWS ParallelCluster 2.x 移至 3.x](#)。

主題

- [定義動作和引數的組態設定](#)
- [引數](#)
- [具有自訂引導操作的範例叢集](#)
- [如何更新 IMDSv2 自訂引導指令碼的範例](#)
- [如何更新 IMDSv1 組態的範例](#)

定義動作和引數的組態設定

下列組態設定用於定義 [HeadNode](#) / [CustomActions](#) / [OnNodeStart](#) & [OnNodeConfigured](#) & [OnNodeUpdated](#) 和 [Scheduling](#) / [OnNodeStart](#) & [CustomActions](#) [OnNodeConfigured](#) 動作和引數。

```
HeadNode:
[...]
```

```
CustomActions:
  OnNodeStart:
    # Script URL. This is run before any of the bootstrap scripts are run
    Script: s3://amzn-s3-demo-bucket/on-node-start.sh
    Args:
      - arg1
  OnNodeConfigured:
    # Script URL. This is run after all the bootstrap scripts are run
    Script: s3://amzn-s3-demo-bucket/on-node-configured.sh
    Args:
      - arg1
  OnNodeUpdated:
    # Script URL. This is run after the head node update is completed.
    Script: s3://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/on-node-start.sh
        Args:
          - arg1
      OnNodeConfigured:
        Script: s3://amzn-s3-demo-bucket/on-node-configured.sh
        Args:
          - arg1
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false
```

使用 Sequence 設定 (在 3.6.0 AWS ParallelCluster 版中新增) :

```
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://amzn-s3-demo-bucket/on-node-start1.sh
          Args:
            - arg1
        - Script: s3://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/on-node-configured1.sh
  Args:
  - arg1
- Script: s3://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/on-node-updated1.sh
  Args:
  - arg1
- Script: s3://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/on-node-start1.sh
          Args:
          - arg1
        - Script: s3://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/on-node-configured1.sh
      Args:
        - arg1
    - Script: s3://amzn-s3-demo-bucket/on-node-configured2.sh
      Args:
        - arg1
    [...]
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false

```

從 3.6.0 AWS ParallelCluster 版開始新增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，而不包含Sequence。

AWS ParallelCluster 不支援同時包含單一指令碼和Sequence相同的自訂動作。例如，如果您指定下列組態，則 AWS ParallelCluster 失敗。

```

[...]
CustomActions:
  OnNodeStart:
    # Script URL. This is run before any of the bootstrap scripts are run
    Script: s3://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/on-node-start1.sh
        Args:
          - arg1
      - Script: s3://amzn-s3-demo-bucket/on-node-start2.sh
        Args:
          - arg1
    [...]

```


引數

In AWS ParallelCluster 2.x 引數是預留的引數，用於存放自訂指令碼的 URL。如果您想要重複使用為 AWS ParallelCluster 2.x with AWS ParallelCluster 3.x 建立的自訂引導指令碼，您需要考慮引數的轉移來調整它們。請參閱 [從 AWS ParallelCluster 2.x 移至 3.x](#)。

具有自訂引導操作的範例叢集

下列步驟會建立簡單的指令碼，以便在設定節點後執行，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至 Amazon S3。如果公有讀取許可不適合您，請使用 [HeadNode / IAM / S3Access](#) 和 [Scheduling / SlurmQueues](#) 組態區段。如需詳細資訊，請參閱 [使用 Amazon S3](#)。

```
$ aws s3 cp --acl public-read /path/to/myscript.sh s3://amzn-s3-demo-bucket/myscript.sh
```

Important

如果在 Windows 上編輯指令碼，則在將指令碼上傳至 Amazon S3 之前，必須將行尾從 CRLF 變更為 LF。

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

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

```
- "wget"
```

如果儲存貯體沒有公有讀取許可，請使用 s3 做為 URL 通訊協定。

```
CustomActions:
  OnNodeConfigured:
    Script: s3://amzn-s3-demo-bucket/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，請登入前端節點，並執行下列命令/var/log/cfn-init.log來檢查位於 cfn-init.log的檔案：

```
$ 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，請檢查cloud-init.log位於運算節點/var/log/cloud-init.log中的。使用 CloudWatch 來檢視這些日誌。

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 CloudWatch Logs 的整合](#)。

如何更新 IMDSv2 自訂引導指令碼的範例

在下列範例中，我們會更新與 IMDSv1 搭配使用的自訂引導操作指令碼，以與 IMDSv2 搭配使用。IMDSv1 指令碼會擷取 Amazon 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 相容之自訂引導操作指令碼。

```
#!/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。

如何更新 IMDSv1 組態的範例

以下是使用 AWS ParallelCluster 3.7.0 及更舊版本時支援 IMDSv1 的叢集組態範例。

```
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 組態中的 [HeadNode](#) // [Iam S3Access](#) 和 [Scheduling / SlurmQueues](#) // [S3Access](#) 參數來設定 AWS ParallelCluster 對 [- Name](#) Amazon S3 [Iam](#) 的存取。

範例

下列範例設定對 *firstbucket/read_only/* 中所有物件的唯讀存取，以及對 *secondbucket/read_and_write/* 中所有物件的讀取/寫入存取。

```
...
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 帳戶中任何儲存貯 *read_only/* 體 (*) 中資料夾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儲存貯體和物件的唯讀存取。

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

使用 競價型執行個體

AWS ParallelCluster 如果您已 SPOT 在叢集組態檔案中將 [SlurmQueues / CapacityType](#) 或 [AwsBatchQueues / CapacityType](#) 設定為 `Spot`，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，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 [Spot 執行個體中斷](#)。若要了解如何 [AwsBatchQueues](#) 使用 Spot 執行個體，請參閱 AWS Batch 《使用者指南》中的 [運算資源](#)。

AWS ParallelCluster 設定的排程器會指派任務來計算具有 Spot 執行個體之佇列中的資源，方式與指派任務來計算具有隨需執行個體之佇列中的資源相同。

使用 Spot 執行個體時，您的帳戶中必須存在 `AWSServiceRoleForEC2Spot` 服務連結角色。若要使用在帳戶中建立此角色 AWS CLI，請執行下列命令：

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

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

下列各節說明使用 `Spot` 執行個體可以中斷的三種情況 [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，如果排程器佇列有待定任務需要額外執行個體，或作用中執行個體的數量低於 [SlurmQueues / ComputeResources /](#)，會 AWS ParallelCluster 嘗試取代執行個體 [MinCount](#)。如果 AWS ParallelCluster 無法佈建新的執行個體，則會定期重複對新執行個體的請求。

案例 2：執行單一節點任務的 Spot 執行個體遭到中斷

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

任務失敗，狀態碼為 `NODE_FAIL`，且任務會重新排入佇列（除非在提交任務時 `--no-requeue` 指定）。如果節點是靜態節點，則會予以取代。如果節點是動態節點，則會終止節點並重設。執行已終止任務的其他節點可能會配置給其他待定任務，或在設定的 [SlurmSettings / ScaledownIdleTime](#) 時間過後縮減規模。

如需 Spot 執行個體的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [Spot 執行個體](#)。

支援的排程器 AWS ParallelCluster

AWS ParallelCluster 支援 Slurm 和 AWS Batch 排程器，這些排程器使用 [Scheduler](#) 設定。下列主題將說明每個排程器及其使用方式。

主題

- [Slurm Workload Manager \(slurm\)](#)
- [搭配使用 AWS Batch \(awsbatch\) 排程器 AWS ParallelCluster](#)

Slurm Workload Manager (**slurm**)

叢集容量大小和更新

叢集的容量是由叢集可以擴展的運算節點數量所定義。運算節點由 AWS ParallelCluster 組態中運算資源內定義的 Amazon EC2 執行個體提供支援 ([Scheduling / SlurmQueues / ComputeResources](#))，並組織成 ([Scheduling / SlurmQueues](#)) 對應 1 : 1 到 Slurm 分割區的佇列。

在運算資源中，您可以設定必須在叢集 () 中持續執行的運算節點（執行個體）數目下限，以及運算資源可擴展至 ([MaxCount3MinCount](#)) 的執行個體數目上限。

在叢集建立時間或叢集更新時，MinCount會針對叢集中定義的每個運算資源 (Scheduling/SlurmQueues/ [ComputeResources](#))，依在 中設定的數目 AWS ParallelCluster 啟動 Amazon EC2 執行個體。啟動以涵蓋叢集中運算資源最小數量節點的執行個體稱為靜態節點。啟動後，除非發生特定事件或條件，否則靜態節點應持續存在於叢集中，系統不會終止這些節點。例如，這類事件包括 Slurm 或 Amazon EC2 運作狀態檢查失敗，以及Slurm節點狀態變更為 DRAIN 或 DOWN。

在 1到 'MaxCount - MinCount'(MaxCount 減去 MinCount)，隨需啟動以處理叢集負載增加的 Amazon EC2 執行個體，稱為動態節點。它們的性質是暫時性的，它們會啟動以提供待定任務，並在它們在叢集組態Scheduling/SlurmSettings/[ScaledownIdleTime](#)中 定義的一段時間內保持閒置後終止（預設值：10 分鐘）。

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，假設有下列 AWS ParallelCluster 組態：

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

下列節點將在 中定義 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]

```

當運算資源具有時 `MinCount == MaxCount`，所有對應的運算節點都會是靜態的，而且所有執行個體都會在叢集建立/更新時間啟動，並保持啟動和執行。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、運算資源或變更運算資源 `MinCount/MaxCount` 的。從 3.9.0 AWS ParallelCluster 版開始，減少佇列的大小需要停止運算機群，或將的 [QueueUpdateStrategy](#) 設定為 `TERMINATE`，才能進行叢集更新。在下列情況下，不需要停止運算機群或將 [QueueUpdateStrategy](#) 設定為 `TERMINATE`：

- 將新佇列新增至排程/[SlurmQueues](#)
- 將新的運算資源 `Scheduling/SlurmQueues/ComputeResources` 新增至佇列
- 增加運算資源 [MaxCount](#) 的
- 增加運算資源的 `MinCount`，並增加相同運算資源至少相同數量的 `MaxCount`

考量和限制

本節旨在概述調整叢集容量時應考慮的任何重要因素、限制或限制。

- 從名稱為 Scheduling/[SlurmQueues](#)的所有運算節點移除佇列時<Queue/Name>-*，靜態和動態都會從Slurm組態中移除，且對應的 Amazon EC2 執行個體將會終止。
- Scheduling/SlurmQueues/[ComputeResources](#) 從佇列移除運算資源時，所有名稱為 的運算節點都會從Slurm組態中移除<Queue/Name>-*-<ComputeResource/Name>-*，且對應的 Amazon EC2 執行個體也會終止。

變更運算資源的 MinCount 參數時，我們可以區分兩種不同的案例：如果 MaxCount 保持等於 MinCount（僅限靜態容量），以及如果 MaxCount 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 MinCount == MaxCount，增加 MinCount（和）MaxCount 時，叢集的設定方式是將靜態節點數量擴展到的新值，MinCount<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>而且系統會繼續嘗試啟動 Amazon EC2 執行個體，以滿足新的必要靜態容量。
- 如果 MinCount == MaxCount，則在減少 MinCount（和 MaxCount）數量 N 時，將透過移除最後一個 N 個靜態節點來設定叢集，<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>]且系統會終止對應的 Amazon EC2 執行個體。
- 初始狀態 MinCount = MaxCount = 100

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

- 在 MinCount 和 -30 上更新 MaxCount: MinCount = 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 數量 N 時（假設 MaxCount 將保持不變），叢集的設定方式是將靜態節點數目擴展到 $\text{MinCount}() \text{ old_MinCount} + N$ 的新值：`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` 且系統會繼續嘗試啟動 Amazon EC2 執行個體，以滿足新的必要靜態容量。此外，為了滿足運算資源的 MaxCount 容量，叢集組態會透過移除最後一個 N 個動態節點來更新：`<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` 且系統會終止對應的 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]
```

- 將 +30 更新為 MinCount ： $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

```
$ 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 $\text{MinCount} < \text{MaxCount}$ ，在增加 MinCount 且數量 MaxCount 相同 N 時，叢集的設定方式是將靜態節點數量擴展到 $\text{MinCount}() \text{ old_MinCount} + N$ 的新值：，`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` 而且系統會繼續嘗試啟動 Amazon EC2 執行個體，以滿足新的必要靜態容量。此外，不會對動態節點數量進行任何變更，以遵守新的

MaxCount 值。

- 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]

```

- 將 +30 更新為 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 $\text{MinCount} < \text{MaxCount}$ ，則在減少 MinCount 數量 N 時（假設 MaxCount 將保持不變），則會透過移除最後一個 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>` 且系統會終止對應的 Amazon EC2 執行個體。此外，為了遵守運算資源的 MaxCount 容量，透過擴展動態節點的數量來填補間隙來更新叢集組態 `MaxCount - new_MinCount: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]`。在此情況下，由於這些是動態節點，除非排程器在新節點上有待定任務，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]

```

- 在上更新 -30 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 $\text{MinCount} < \text{MaxCount}$ ，則在減少 MinCount 和相同數量 MaxCount 的 N 時，將透過移除最後一個 N 個靜態節點來設定叢集，`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] 且系統會終止對應的 Amazon EC2 執行個體。

此外，不會對動態節點數量進行任何變更，以遵守新 MaxCount 值。

- 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]
```

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

```
$ 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 $\text{MinCount} < \text{MaxCount}$ ，則在減少 MaxCount 數量 N 時（假設 MinCount 將保持不變），叢集將透過移除最後一個 N 個動態節點進行設定，`<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N>...<oldMaxCount>`] 如果靜態節點受到 `running.No` 影響，系統會終止對應的 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]
```

- 在上更新 -30 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]
```

對任務的影響

在移除節點並終止 Amazon EC2 執行個體的所有情況下，除非沒有其他節點滿足任務需求，否則在移除的節點上執行的 sbatch 任務會重新排入佇列。在此情況下，任務會失敗，狀態為 NODE_FAIL，並從佇列中消失，且必須手動重新提交。

如果您打算執行叢集調整大小更新，您可以防止任務在計劃更新期間將移除的節點中執行。這可以透過設定要在維護中移除的節點來實現。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任務。

假設透過計劃的叢集調整大小更新，您將移除節點 `queue-st-computeresource-[9-10]`。您可以使用下列命令建立 Slurm 保留

```
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]
```

這將在節點 `maint_for_update` 上建立名為 `queue-st-computeresource-[9-10]` 的 Slurm 保留。從建立保留開始，就無法再將任何任務執行到節點 `queue-st-computeresource-[9-10]`。請注意，保留不會阻止任務最終配置到節點 `queue-st-computeresource-[9-10]`。

叢集調整大小更新後，如果僅在調整大小更新期間移除的節點上設定 Slurm 保留，系統會自動刪除維護保留。如果您改為在叢集調整大小更新後仍存在的節點上建立 Slurm 保留，我們可能會想要在執行調整大小更新後移除節點上的維護保留，方法是使用下列命令

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

如需 Slurm 保留的其他詳細資訊，請參閱 [此處](#) 的官方 SchedMD 文件。

容量變更的叢集更新程序

在排程器組態變更時，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 AWS ParallelCluster 組態產生更新的Slurm分割區組態
- 重新啟動 slurmctld (透過 Chef 服務配方完成)
- 檢查slurmctld狀態 (`systemctl is-active --quiet slurmctld.service`)
- 重新載入Slurm組態 (`scontrol reconfigure`)
- 啟動 clustermgtd (`supervisorctl start clustermgtd`)

如需的詳細資訊Slurm，請參閱 <https://slurm.schedmd.com>。如需下載，請參閱 <https://github.com/SchedMD/slurm/tags>。如需來源碼，請參閱 <https://github.com/SchedMD/slurm>。

支援的叢集和 SLURM 版本

下表列出 AWS 支援的 AWS ParallelCluster 和 Slurm版本。

AWS ParallelCluster version(s)	支援的 Slurm 版本
3.13.0	24.05.07
3.12.0	23.11.10
3.11.0	23.11.10
3.9.2、3.9.3、3.10.0	23.11.7
3.9.0、3.9.1	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

AWS ParallelCluster version(s)	支援的 Slurm 版本
3.6.0、3.6.1	23.02.2
3.5.0、3.5.1	22.05.8
3.4.0、3.4.1	22.05.7
3.3.0、3.3.1	22.05.5
3.1.4、3.1.5、3.2.0、3.2.1	21.08.8-2
3.1.2、3.1.3	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 適用於多個佇列模式的 指南](#)
- [Slurm 叢集保護模式](#)
- [Slurm 叢集快速容量不足容錯移轉](#)
- [Slurm 記憶體型排程](#)
- [使用 Slurm 進行多個執行個體類型配置](#)
- [動態節點的叢集擴展](#)
- [Slurm 使用 會計 AWS ParallelCluster](#)
- [Slurm 組態自訂](#)
- [Slurm 與 prologepilog](#)
- [叢集容量大小和更新](#)

多個佇列的組態

使用第 3 AWS ParallelCluster 版，您可以將 [Scheduler](#) 設定為 `slurm`並在組態檔案中為 指定多個佇列 [SlurmQueues](#)，以設定多個佇列。在此模式中，組態檔案 [ComputeResources](#)區段中指定的運

算節點中共存不同的執行個體類型。[ComputeResources](#)具有不同執行個體類型的 會視需要擴展或縮減[SlurmQueues](#)。

當工作負載共用相同的基礎基礎設施和資源（例如共用儲存、聯網或登入節點）時，單一叢集內的多個佇列通常優於多個叢集。如果工作負載有類似的運算、儲存和聯網需求，在單一叢集中使用多個佇列會更有效率，因為它允許資源共用並避免不必要的重複。這種方法簡化了管理並降低了額外負荷，同時仍然允許有效率的任務排程和資源配置。另一方面，當工作負載之間有強大的安全性、資料或操作隔離需求時，應使用多個叢集。例如，如果您需要獨立管理和操作工作負載，使用不同的排程、更新週期或存取政策，則多個叢集更為合適。

叢集佇列和運算資源配額

資源	配額
Slurm queues	每個叢集 50 個佇列
Compute resources	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[ComputeResources](#)中每個運算資源都必須具有唯一的 [Name](#)、[MinCount](#)、[InstanceType](#)和 [MaxCount](#)。[MinCount](#) [MaxCount](#)具有預設值，可定義[ComputeResources](#)佇列中運算資源的執行個體範圍。您也可以為 [MinCount](#)和 指定自己的值[MaxCount](#)。中的每個運算資源[ComputeResources](#)是由編號為 1 到的靜態節點，[MinCount](#)以及編號為 [MinCount](#)到 值的動態節點所組成[MaxCount](#)。

範例組態

以下是叢集組態檔案的排程區段範例。在此組態中，有兩個名為 queue1和 的佇列，queue2每個佇列[ComputeResources](#)都有指定的 [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

在運算機群中啟動的執行個體會動態指派。系統會為每個節點產生主機名稱。根據預設，AWS ParallelCluster 會使用下列格式的主機名稱：

```
$HOSTNAME=$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」為「queue-name」。
- \$STATDYN st 適用於靜態節點，或 dy 適用於動態節點。
- \$COMPUTE_RESOURCE 是與此節點對應的 [ComputeResources](#) 運算資源 [Name](#) 的。
- \$NODENUM 是節點的數目。[MinCount](#) 靜態節點的值 \$NODENUM 介於 - (1) 和 - 之間，動態節點的值介於 - (1) 和 [MaxCount-MinCount](#) 之間。

從上述範例組態檔案中，來自 queue1 和 運算資源的指定節點 c5xlarge 具有主機名稱：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。FQDN 是 \$HOSTNAME.\$CLUSTERNAME.pcluster，其中 \$CLUSTERNAME 是叢集的名稱。

請注意，相同的格式也會用於 Slurm 節點名稱。

使用者可以選擇使用支援運算節點之執行個體的預設 Amazon EC2 主機名稱，而非使用的預設主機名稱格式 AWS ParallelCluster。這可以透過將 [UseEc2Hostnames](#) 參數設定為 true 來完成。不過，Slurm 節點名稱將繼續使用預設 AWS ParallelCluster 格式。

Slurm 適用於多個佇列模式的 指南

在這裡，您可以了解如何 AWS ParallelCluster Slurm 和管理佇列（分割區）節點，以及如何監控佇列和節點狀態。

概觀

擴展架構是以 Slurm的[雲端排程指南](#)和省電外掛程式為基礎。如需省電外掛程式的詳細資訊，請參閱[Slurm省電指南](#)。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。

雲端節點生命週期

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

- 狀態中的節點**POWER_SAVING**會在 中顯示~尾碼 (例如 idle~)sinfo。在此狀態下，沒有 EC2 執行個體正在備份節點。不過，仍然Slurm可以將任務配置給節點。
- 轉換為 **POWER_UP** 狀態的節點會在 中顯示#尾碼 (例如 idle#)sinfo。當將任務Slurm配置到 POWER_UP 狀態的節點時，節點會自動轉換為 POWER_SAVING 狀態。

或者，您可以使用 命令，以su根使用者身分手動將節點轉換為 POWER_UP 狀態：

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

在此階段中，會ResumeProgram叫用、啟動和設定 EC2 執行個體，以及節點轉換為 POWER_UP 狀態。

- 目前可供使用的節點會在 中顯示沒有尾碼 (例如 idle)sinfo。節點設定完成並加入叢集後，即可執行任務。在此階段中，節點已正確設定並可供使用。

一般而言，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，靜態節點會在建立叢集後可用。

- 轉換為**POWER_DOWN**狀態的節點會在 中顯示%尾碼 (例如 idle%)sinfo。動態節點會在之後自動進入 POWER_DOWN 狀態[ScaledownIdletime](#)。相反地，在大多數情況下，靜態節點不會關閉電源。不過，您可以使用 命令，以su根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，而節點會設回 POWER_SAVING 狀態，並在之後使用[ScaledownIdletime](#)。

[ScaledownIdletime](#) 設定會儲存至Slurm組態設定SuspendTimeout。

- 離線的節點會在中顯示*尾碼 (例如 down*)sinfo。如果Slurm控制器無法聯絡節點，或靜態節點已停用且備份執行個體終止，節點就會離線。

考慮以下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 POWER_UP 狀態，應該可在幾分鐘內使用。gpu-dy-gpucompute1-1 節點處於 POWER_DOWN 狀態，並在 [ScaledownIdleTime](#) (預設為 10 分鐘) 之後轉換為 POWER_SAVING 狀態。

所有其他節點都處於 POWER_SAVING 狀態，沒有 EC2 執行個體支援這些節點。

使用可用的節點

可用的節點由 Amazon EC2 執行個體支援。根據預設，節點名稱可用來直接將 SSH 傳入執行個體 (例如 `ssh efa-st-efacompute1-1`)。您可以使用 `命令` 擷取執行個體的私有 IP 地址：

```
$ scontrol show nodes nodename
```

檢查傳回 NodeAddr 欄位中的 IP 地址。

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。相反地，它應該與節點名稱相同。

任務狀態和提交

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

如果為任務配置的節點包含任何處於 POWER_SAVING 狀態的節點，任務會以 CF、或 CONFIGURING 狀態開始。此時，任務會等待 POWER_SAVING 處於 狀態的節點轉換為 POWER_UP 狀態並變成可用。

在為任務配置的所有節點都可用之後，任務會進入 RUNNING(R) 狀態。

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

所有節點都設定了下列功能，可用於任務提交命令：

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

您可以使用 命令來查看特定節點的功能：

```
$ scontrol show nodes nodename
```

在傳回中，檢查 AvailableFeatures 清單。

考慮叢集的初始狀態，您可以透過執行 sinfo 命令來檢視。

```
$ 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，spot 是預設佇列。它以*尾碼表示。

將任務提交至預設佇列中的一個靜態節點 (spot)。

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

將任務提交至EFA佇列中的一個動態節點。

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

將任務提交至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]"
```

將任務提交至gpu佇列中的一個 GPU 節點。

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

使用 squeue命令來考慮任務的狀態。

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
12 ondemand wrap ubuntu CF 0:36 10 ondemand-dy-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
```

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

```
# Nodes states corresponds to state of running jobs
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
efa up infinite 3 idle~ efa-dy-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，節點狀態由 AWS ParallelCluster 依據本主題稍早所述的雲端節點生命週期中的特定程序進行完整管理。

不過，AWS ParallelCluster 也會取代或終止 DOWN 和 DRAINED 狀態中運作狀態不佳的節點，以及具有運作狀態不佳後端執行個體的節點。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。Slurm 分割區是其中的佇列 AWS ParallelCluster。

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

pcluster update-compute-fleet

- 停止運算機群 - 執行下列命令時，所有分割區都會轉換為 INACTIVE 狀態，而 AWS ParallelCluster 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
```

- 啟動運算機群 - 執行下列命令時，所有分割區一開始都會轉換為 UP 狀態。不過，AWS ParallelCluster 程序不會將分割區保持在 UP 狀態。您需要手動變更分割區狀態。所有靜態節點會在幾分鐘後可用。請注意，將分割區設定為 UP 不會啟動任何動態容量。

```
$ 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，以檢查叢集的狀態 Status。下列列出可能的狀態：

- STOP_REQUESTED：停止運算機群請求會傳送至叢集。
- STOPPING：pcluster 程序目前正在停止運算機群。

- **STOPPED** : pcluster程序已完成停止程序、所有分割區都處於 INACTIVE 狀態，且所有運算執行個體都會終止。
- **START_REQUESTED** : 啟動運算機群請求會傳送至叢集。
- **STARTING** : pcluster程序目前正在啟動叢集。
- **RUNNING** : pcluster程序已完成啟動程序、所有分割區都處於 UP 狀態，且幾分鐘後可使用靜態節點。
- **PROTECTED** : 此狀態表示某些分割區具有一致的引導失敗。受影響的分割區處於非作用中狀態。請調查問題，然後執行 `update-compute-fleet`以重新啟用機群。

手動控制佇列

在某些情況下，您可能想要對叢集中的節點或佇列（在中稱為分割區Slurm）進行一些手動控制。您可以使用 `scontrol`命令，透過下列常見程序來管理叢集中的節點。

- 啟動處於 **POWER_SAVING** 狀態的動態節點

以su根使用者身分執行 命令：

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

您也可以提交請求特定數量節點的預留位置sleep 1任務，然後依賴 Slurm 來啟動所需的節點數量。

- 在之前關閉動態節點 [ScaledownIdletime](#)

我們建議您使用 命令將動態節點設定為 DOWN作為su根使用者：

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

AWS ParallelCluster 會自動終止和重設下降的動態節點。

一般而言，我們不建議您POWER_DOWN直接使用 `scontrol update nodename=nodename state=power_down`命令將節點設定為。這是因為 AWS ParallelCluster 會自動處理關機程序。

- 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 命令將特定佇列設定為 INACTIVE做為su根使用者：

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

這樣做會終止分割區中的所有執行個體備份節點。

- 啟用佇列 (分割區)

使用 命令UP將特定佇列設定為su根使用者：

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

擴展行為和調整

以下是正常擴展工作流程的範例：

- 排程器會收到需要兩個節點的任務。
- 排程器會將兩個節點轉換為 POWER_UP 狀態，並使用ResumeProgram節點名稱呼叫 (例如 queue1-dy-spotcompute1-[1-2])。
- ResumeProgram 會啟動兩個 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 30 分鐘，然後再重設節點。
- 已設定執行個體並加入叢集。任務開始在執行個體上執行。
- 任務完成並停止執行。
- 經過設定的 SuspendTime (設定為 [ScaledownIdleTime](#)) 之後，排程器會將執行個體設定為 POWER_SAVING 狀態。排程器接著會queue1-dy-spotcompute1-[1-2]設定為 POWER_DOWN 狀態SuspendProgram，並使用節點名稱呼叫。
- SuspendProgram 會針對兩個節點呼叫。節點會保持 POWER_DOWN 狀態，例如，保留 idle% SuspendTimeout (預設期間為 120 秒 (2 分鐘))。在 clustermgtd偵測到節點正在關閉電源後，它會終止備份執行個體。然後，它會轉換為queue1-dy-spotcompute1-[1-2]閒置狀態，並重設私有 IP 地址和主機名稱，以便為未來的任務做好準備。

如果發生錯誤，且特定節點的執行個體因為某些原因而無法啟動，則會發生下列情況：

- 排程器會收到需要兩個節點的任務。
- 排程器會將兩個雲端爆量節點轉換為 POWER_UP 狀態ResumeProgram，並使用節點名稱呼叫 (例如 queue1-dy-spotcompute1-[1-2])。
- ResumeProgram 只會啟動一 (1) 個 Amazon EC2 執行個體queue1-dy-spotcompute1-1，並設定，其中一 (1) 個執行個體 queue1-dy-spotcompute1-2無法啟動。
- queue1-dy-spotcompute1-1 不會受到影響，並在達到 POWER_UP 狀態後上線。

- `queue1-dy-spotcompute1-2` 會轉換為 `POWER_DOWN` 狀態，且任務會自動排入佇列，因為 Slurm 偵測到節點故障。
- `queue1-dy-spotcompute1-2` 會在 `SuspendTimeout` (預設值為 120 秒 (2 分鐘)) 之後提供。同時，任務會重新排入佇列，並且可以在另一個節點上開始執行。
- 上述程序會重複執行，直到任務可以在可用的節點上執行，而不會發生失敗。

如有需要，可以調整兩個計時參數：

- **ResumeTimeout** (預設值為 30 分鐘)：ResumeTimeout 控制節點轉換為停機狀態之前 Slurm 等待的時間。
 - ResumeTimeout 如果您的安裝前/後程序需要近乎這麼長的時間，擴展可能很有用。
 - ResumeTimeout 如果發生問題，也是取代或重設節點之前 AWS ParallelCluster 等待的時間上限。如果在啟動或設定期間發生任何錯誤，運算節點會自行終止。會在偵測到已終止的執行個體時 AWS ParallelCluster 處理取代節點。
- **SuspendTimeout** (預設值為 120 秒 (2 分鐘))：SuspendTimeout 控制節點放回系統並準備好再次使用的速度。
 - 較短 SuspendTimeout 表示節點重設速度更快，並且 Slurm 可以嘗試更頻繁地啟動執行個體。
 - 較長 SuspendTimeout 表示失敗的節點重設速度較慢。同時，Slurm 會嘗試使用其他節點。如果 SuspendTimeout 超過幾分鐘，Slurm 會嘗試循環系統中的所有節點。較長 SuspendTimeout 的時間可能有利於大規模系統 (超過 1,000 個節點)，以減少其嘗試頻繁重新排入佇列失敗任務 Slurm 時的壓力。
 - 請注意，SuspendTimeout 不是指 AWS ParallelCluster 等待終止節點備份執行個體的時間。POWER_DOWN 節點的備份執行個體會立即終止。終止程序通常會在幾分鐘內完成。不過，在此期間，節點會保持 POWER_DOWN 狀態，且不適用於排程器。

架構的日誌

下列清單包含金鑰日誌。與 Amazon CloudWatch Logs 搭配使用的日誌串流名稱格式為 `{hostname}.{instance_id}.{logIdentifier}`，其中 `logIdentifier` 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)`

常見問題以及如何除錯：

無法啟動、開啟電源或加入叢集的節點

- 動態節點：
 - 檢查ResumeProgram日誌，查看ResumeProgram是否已使用節點呼叫。如果沒有，請檢查slurmctld日誌以判斷是否Slurm嘗試ResumeProgram使用節點呼叫。請注意，上的許可不正確ResumeProgram可能會導致其無提示失敗。
 - 如果呼叫ResumeProgram，請檢查是否已為節點啟動執行個體。如果執行個體未啟動，應該會出現清楚的錯誤訊息，指出執行個體無法啟動的原因。
 - 如果執行個體已啟動，在引導程序期間可能有些問題。從ResumeProgram日誌尋找對應的私有IP地址和執行個體ID，並在CloudWatch Logs中查看特定執行個體的對應引導日誌。
- 靜態節點：
 - 檢查clustermgtd日誌，查看是否已為節點啟動執行個體。如果執行個體未啟動，執行個體無法啟動的原因應該會發生明確的錯誤。
 - 如果執行個體已啟動，引導程序會有些問題。從clustermgtd日誌尋找對應的私有IP和執行個體ID，並查看CloudWatch Logs中特定執行個體的對應引導日誌。

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。若要檢查是否已clustermgtd取代或終止節點，請檢查clustermgtd日誌。
 - 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，節點為DOWN），請檢查slurmctld日誌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SDKs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未終止節點，請檢查是否已computemgtd終止節點，或EC2是否已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，如果節點失敗，任務會自動重新排入佇列。在slurmctld日誌中查看任務或節點失敗的原因，並從那裡評估情況。

更換或終止執行個體時失敗，關閉節點時失敗

- 一般而言，clustermgtd會處理所有預期的執行個體終止動作。在clustermgtd日誌中查看無法取代或終止節點的原因。
- 對於故障的動態節點[ScaledownIdleTime](#)，請查看SuspendProgram日誌中的slurmctld程序是否使用特定節點做為引數進行呼叫。請注意，實際上SuspendProgram不會執行任何特定動作。相反地，它只會在呼叫時記錄。所有執行個體終止和NodeAddr重設都會由完成clustermgtd。會在IDLE之後將節點Slurm轉換為SuspendTimeout。

其他問題：

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

如需有關任務配置、節點配置和擴展決策的問題，請查看slurmctld日誌是否有錯誤。

Slurm 叢集保護模式

當叢集在啟用受保護模式的情況下執行時，AWS ParallelCluster 監控並追蹤運算節點啟動時的運算節點引導失敗。它這樣做是為了偵測這些失敗是否持續發生。

如果在佇列（分割區）中偵測到下列項目，則叢集會進入受保護狀態：

1. 連續運算節點引導失敗會持續發生，而不會成功啟動運算節點。
2. 失敗計數達到預先定義的閾值。

叢集進入受保護狀態後，會 AWS ParallelCluster 停用故障等於或高於預先定義閾值的佇列。

Slurm 3.0.0 AWS ParallelCluster 版中已新增 叢集保護模式。

您可以使用受保護模式來減少運算節點引導失敗循環所花費的時間和資源。

受保護模式參數

protected_failure_count

`protected_failure_count` 指定在佇列（分割區）中啟用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 `protected_failure_count` 大於零，則會啟用受保護模式。

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

您可以在位於的 `clustermgtd` 組態檔案中新增參數來變更 `protected_failure_count` 值 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` `HeadNode`。

您可以隨時更新此參數，而且您不需要停止運算機群即可執行此操作。如果啟動在失敗計數達到之前，在佇列中成功 `protected_failure_count`，則失敗計數會重設為零。

檢查處於受保護狀態的叢集狀態

當叢集處於受保護狀態時，您可以檢查運算機群狀態和節點狀態。

運算機群狀態

運算機群的狀態位於處於受保護狀態的叢集 `PROTECTED` 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

若要了解哪些佇列（分割區）的引導失敗已啟用受保護狀態，請登入叢集並執行 `sinfo` 命令。開機失敗等於或高於的分割區 `protected_failure_count` 處於 `INACTIVE` 狀態。在或以上沒有引導失敗的分割區 `protected_failure_count` 處於 `UP` 狀態並如預期運作。

`PROTECTED` 狀態不會影響執行中的任務。如果任務在開機失敗位於或高於的分割區上執行 `protected_failure_count`，則在執行中的任務完成後 `INACTIVE`，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*  inactive infinite 10 down% queue1-dy-c5xlarge-[1-10]
queue1*  inactive infinite 3490 idle~ queue1-dy-c5xlarge-[11-3500]
queue2  up  infinite 10 idle~ queue2-dy-c5xlarge-[1-10]
```

分割區queue1是INACTIVE由於偵測到連續 10 個運算節點引導失敗。

節點後方的執行個體queue1-dy-c5xlarge-[1-10]已啟動，但由於運作狀態不佳而無法加入叢集。叢集處於受保護狀態。

分割區queue2不會受到 中引導失敗的影響queue1。它處於 UP 狀態，仍然可以執行任務。

如何停用受保護狀態

解決引導錯誤之後，您可以執行下列命令，將叢集移出受保護的狀態。

```
$ 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

啟用受保護狀態的引導錯誤會細分為以下三種類型。若要識別類型和問題，您可以檢查 AWS ParallelCluster 產生的日誌。如果日誌已產生，您可以檢查日誌是否有錯誤詳細資訊。如需詳細資訊，請參閱[擷取和保留日誌](#)。

1. 引導錯誤，導致執行個體自行終止。

執行個體在引導程序初期失敗，例如因為 [CustomActions](#) \ [OnNodeStart](#) | [SlurmQueues](#) \ [OnNodeConfigured](#)指令碼中的錯誤而自行終止的執行個體。

對於動態節點，請尋找類似下列的錯誤：

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

對於靜態節點，請查看clustermgtd日誌 (/var/log/parallelcluster/clustermgtd) 中是否有類似以下的錯誤：

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

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

執行個體無法連結 `resume_timeout` (適用於動態節點) 或 `node_replacement_timeout` (適用於靜態節點) 中的叢集。在逾時之前不會自行終止。例如，叢集的聯網設定不正確，且節點在逾時到期Slurm後設定為 DOWN 狀態。

對於動態節點，請尋找類似下列的錯誤：

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

對於靜態節點，請查看 `clustermgtd` 日誌 (`/var/log/parallelcluster/clustermgtd`) 中是否有類似以下的錯誤：

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

3. 節點運作狀態檢查失敗。

節點後方的執行個體未通過 Amazon EC2 運作狀態檢查或排程事件運作狀態檢查，且節點視為引導失敗節點。在此情況下，執行個體會因無法控制的原因而終止 AWS ParallelCluster。

查看 `clustermgtd` 日誌 (`/var/log/parallelcluster/clustermgtd`) 中是否有類似以下的錯誤：

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

4. 運算節點註冊失敗Slurm。

向Slurm控制 `slurmd` 常駐程式 (`slurmctld`) 註冊常駐程式失敗，並導致運算節點狀態變更為 `INVALID_REG` 狀態。設定錯誤的Slurm運算節點可能會導致此錯誤，例如使用 [CustomSlurmSettings](#) 運算節點規格錯誤設定的運算節點。

查看前端節點上的 `slurmctld` 日誌檔案 (`/var/log/slurmctld.log`)，或查看故障運算節點的 `slurmd` 日誌檔案 (`/var/log/slurmd.log`) 是否有類似以下的錯誤：

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

如何偵錯受保護模式

如果您的叢集處於受保護狀態，而且從 AWS ParallelCluster 產生的 `clustermgtd` 日誌 `HeadNode` 和有問題的運算節點產生的 `cloud-init-output` 日誌，則可以檢查日誌以取得錯誤詳細資訊。如需如何擷取日誌的詳細資訊，請參閱 [擷取和保留日誌](#)。

`clustermgtd` 前端節點上的 `log(/var/log/parallelcluster/clustermgtd)`

日誌訊息會顯示哪些分割區發生引導失敗，以及對應的引導失敗計數。

```
[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` 日誌中，搜尋 `Found the following bootstrap failure nodes` 以尋找引導失敗的節點。

```
[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 `clustermgtd` 日誌中，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 `clustermgtd` 日誌中取得引導失敗節點私有 IP 地址後，您可以登入運算節點或遵循 [中的指引擷取和保留日誌](#) 來擷取日誌，以尋找對應的運算節點日誌。在大多數情況下，來自有問題節點的 `/var/log/cloud-init-output` 日誌會顯示導致運算節點引導失敗的步驟。

Slurm 叢集快速容量不足容錯移轉

從 3.2.0 AWS ParallelCluster 版開始，叢集執行時預設會啟用快速容量不足容錯移轉模式。這可將偵測到 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。當您使用使用不同執行個體類型的多個運算資源來設定佇列時，這特別有效。

Amazon EC2 偵測到容量不足故障：

- `InsufficientInstanceCapacity`
- `InsufficientHostCapacity`
- `InsufficientReservedInstanceCapacity`
- `MaxSpotInstanceCountExceeded`
- `SpotMaxPriceTooLow`：如果 Spot 請求價格低於所需的 Spot 請求履行價格下限，則會啟用。
- `Unsupported`：使用特定中不支援的執行個體類型來啟用 AWS 區域。

在容量快速不足容錯移轉模式下，如果在將任務指派給 [SlurmQueues](#) / 時偵測到容量不足錯誤 [compute resource](#)，會 AWS ParallelCluster 執行下列動作：

1. 它會在預先定義的期間內將運算資源設定為停用 (DOWN) 狀態。
2. 它使用 `POWER_DOWN_FORCE` 來取消運算資源失敗的節點任務，並暫停失敗的節點。它會將失敗的節點設定為 `IDLE` 和 `POWER_DOWN (!)` 狀態，然後設定為 `POWERING_DOWN (%)`。
3. 它會將任務排入另一個運算資源的佇列。

停用之運算資源的靜態和已啟動節點不會受到影響。任務可以在這些節點上完成。

此週期會重複，直到任務成功指派給運算資源節點。如需節點狀態的資訊，請參閱 [Slurm 適用於多個佇列模式的指南](#)。

如果找不到執行任務的運算資源，任務會設定為 `PENDING` 狀態，直到經過預先定義的期間為止。在此情況下，您可以修改預先定義的時段，如下節所述。

容量不足逾時參數

`insufficient_capacity_timeout`

`insufficient_capacity_timeout` 指定當偵測到容量不足錯誤時，運算資源保持在停用 (down) 狀態的期間（以秒為單位）。

預設 `insufficient_capacity_timeout` 會啟用。

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

如果 `insufficient_capacity_timeout` 值小於或等於零，則會停用快速容量不足容錯移轉模式。

您可以在位於的 `clustermgtd` 組態檔案中新增 參數 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` 來變更 `insufficient_capacity_timeout` 值 `HeadNode`。

您可以隨時更新 參數，而無需停止運算機群。

例如：

- `insufficient_capacity_timeout=600`:

如果偵測到容量不足錯誤，則運算資源會設為已停用 (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，表示運算資源已停用 (DOWN)。1 分鐘後，其失敗的節點會設定為 `idle~` 狀態。

- `insufficient_capacity_timeout=0`:

容量不足快速容錯移轉模式已停用。運算資源不會停用。

Note

當節點故障時，容量不足錯誤，以及叢集管理協助程式偵測到節點故障的時間之間，可能會有最多一分鐘的延遲。這是因為叢集管理協助程式會檢查節點容量不足失敗，並以一分鐘的間隔將運算資源設定為 `down` 狀態。

容量快速不足容錯移轉模式狀態

當叢集處於快速容量不足容錯移轉模式時，您可以檢查其狀態和節點狀態。

節點狀態

當任務提交至運算資源動態節點，並偵測到容量不足錯誤時，節點會處於 `down#` 狀態並具有原因。

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

然後，關閉電源的節點（處於 `idle~` 狀態的節點）會設定為 `down~` 並說明原因。

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

任務會重新排入佇列中其他運算資源的佇列。

運算資源靜態節點和UP不受快速容量不足容錯移轉模式影響的節點。

請考慮以下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]
```

我們會向需要一個節點的 queue1 提交任務。

```
$ 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]
```

queue1-dy-c-1-1 啟動節點以執行任務。不過，執行個體因為容量不足錯誤而無法啟動。節點 queue1-dy-c-1-1 設定為 down。運算資源 (queue2-dy-c-1) 中的關閉電源動態節點設定為 down。

您可以使用 `scontrol show nodes` 檢查節點原因。

```
$ 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]
```

任務會排入佇列運算資源中的另一個執行個體類型佇列。

`insufficient_capacity_timeout` 經過之後，運算資源中的節點會重設為 `idle~` 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]
```

運算資源中的經過 `insufficient_capacity_timeout` 和節點重設為 `idle~` 狀態後，Slurm 排程器會將節點的優先順序較低。除非發生下列其中一種情況，否則排程器會持續從權重較高的其他佇列運算資源中選取節點：

- 任務的提交需求符合復原的運算資源。
- 沒有其他運算資源可用，因為它們具有容量。
- `slurmctld` 會重新啟動。
- AWS ParallelCluster 運算機群會停止並開始關閉電源並開啟所有節點的電源。

相關日誌

與容量不足錯誤和快速容量不足容錯移轉模式相關的日誌，可在 Slurm 的 `resume` 日誌中找到，並在前端節點中 `clustermgtd` 登入。

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

當節點因為容量不足而無法啟動時的錯誤訊息。

```
[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`)

由於容量不足，Queue1 中的運算資源 c-1 已停用。

```
[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
```

容量不足逾時過期後，運算資源會重設，運算資源內的節點會設定為 `idle~`。

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the following compute resources because insufficient capacity timeout expired: {'queue1': ['c-1']}
```

Slurm 記憶體型排程

從 3.2.0 版開始，AWS ParallelCluster 支援使用 / [SlurmSettings](#) [EnableMemoryBasedScheduling](#) 叢集組態參數的 Slurm 記憶體型排程。

Note

從 3.7.0 AWS ParallelCluster 版開始，如果您在 [執行個體](#) 中設定多個執行個體類型，`EnableMemoryBasedScheduling` 則可以啟用。
對於 3.2.0 到 3.6.x AWS ParallelCluster 版，如果您在 [執行個體](#) 中設定多個執行個體類型，則 `EnableMemoryBasedScheduling` 無法啟用。

Warning

當您在 `EnableMemoryBasedScheduling` 啟用的 Slurm 佇列運算資源中指定多個執行個體類型時，該 `RealMemory` 值是可供所有執行個體類型使用的記憶體數量下限。如果您指定具有非常不同記憶體容量的執行個體類型，這可能會導致大量未使用的記憶體。

使用 `EnableMemoryBasedScheduling: true`，Slurm 排程器會追蹤每個任務在每個節點上所需的記憶體量。然後，Slurm 排程器會使用此資訊來排程相同運算節點上的多個任務。任務在節點上所需的記憶體總量不能大於可用的節點記憶體。排程器可防止任務使用比提交任務時請求更多的記憶體。

使用 `EnableMemoryBasedScheduling: false`，任務可能會爭奪共用節點上的記憶體，並導致任務失敗和 `out-of-memory` 事件。

Warning

Slurm 為其標籤使用 2 個表示法，例如 MB 或 GB。分別將這些標籤讀取為 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 中的消耗性資源 Slurm。
- [ConstrainRAMSpace=yes](#) 中的 Slurm `cgrouop.conf`。使用此選項，任務對記憶體的存取僅限於提交任務時請求的記憶體量。

Note

設定這兩個選項時，其他幾個 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)

根據預設，`EnableMemoryBasedScheduling` 設定為 `false`。當 `false` 時，Slurm 不會在其排程演算法中包含記憶體做為資源，也不會追蹤任務使用的記憶體。使用者可以指定 `--mem MEM_PER_NODE` 選項，以設定任務所需的每個節點的記憶體數量下限。這會強制排程器在排程任務 `MEM_PER_NODE` 時選擇 `RealMemory` 值至少為 的節點。

例如，假設使用者使用 提交兩個任務 `--mem=5GB`。如果 CPUs 或 GPUs 等請求的資源可用，任務可以同時在具有 8 GiB 記憶體的節點上執行。這兩個任務不會排程在小於 5 GiB 的運算節點上 `RealMemory`。

⚠ Warning

停用記憶體型排程時，Slurm不會追蹤任務使用的記憶體量。在相同節點上執行的任務可能會競爭記憶體資源，並導致其他任務失敗。

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與 [Slurm 文件](#) 中所述行為不同的行為。

EnableMemoryBasedScheduling: true

當 `EnableMemoryBasedScheduling` 設為 `true` 時，會Slurm追蹤每個任務的記憶體用量，並防止任務使用比 `--mem` 提交選項請求更多的記憶體。

使用上一個範例，使用者使用 提交兩個任務 `--mem=5GB`。任務無法在具有 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，`--mem-per-cpu` 並與Slurm文件中描述 `--mem-per-gpu` 的內容一致。例如，使用 提交任務 `--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB`。在此情況下，會為每個節點Slurm指派總計 4 GiB 的任務。

⚠ Warning

啟用記憶體型排程時，我們建議使用者在提交任務時包含 `--mem` 規格。使用隨附的預設Slurm組態 AWS ParallelCluster，如果未包含記憶體選項 (`--mem`、或 `--mem-per-gpu`) `--mem-per-cpu`，會將已配置節點的整個記憶體Slurm指派給任務，即使它只請求一部分的其他資源，例如 CPUs 或 GPUs。這樣可以有效地防止節點共用，直到任務完成為止，因為其他任務沒有可用的記憶體。這是因為當任務提交時未提供記憶體規格 `DefMemPerNode` 時，Slurm會將任務的每個節點的記憶體設定為 `0`。此參數的預設值為 `0`，並指定對節點記憶體的無限制存取。

如果具有不同記憶體數量的多種運算資源可在相同佇列中使用，則提交而沒有記憶體選項的任務可能會在不同的節點上指派不同數量的記憶體。這取決於排程器提供給任務的節點。使用者可以在Slurm組態檔案中的叢集或分割區層級定義選項的自訂值 `DefMemPerCPU`，例如 `DefMemPerNode` 或 `DefMemPerGPU`，以防止此行為。

Slurm RealMemory 和 AWS ParallelCluster SchedulableMemory

使用隨附的Slurm組態 AWS ParallelCluster，Slurm 會將 [RealMemory](#) 解譯為每個節點可供任務使用的記憶體量。從 3.2.0 版開始，預設會將 RealMemoryAmazon [Amazon EC2 執行個體類型](#) 中列出的記憶體 AWS ParallelCluster 設定為 95%，並由 Amazon EC2 API [DescribeInstanceTypes](#) 傳回。

停用記憶體型排程時，Slurm排程器會在使用者使用 `--mem` 指定的 提交任務時RealMemory，使用 來篩選節點。

啟用記憶體型排程時，Slurm排程器會將 解譯RealMemory為運算節點上執行之任務可用的記憶體數量上限。

預設設定可能不適用於所有執行個體類型：

- 此設定可能高於節點實際可存取的記憶體數量。當運算節點是小型執行個體類型時，可能會發生這種情況。
- 此設定可能低於節點實際可存取的記憶體數量。當運算節點是大型執行個體類型，並可能導致大量未使用的記憶體時，就會發生這種情況。

您可以使用 [SlurmQueues / ComputeResources / SchedulableMemory](#)來微調 AWS ParallelCluster 為運算節點RealMemory設定的 值。若要覆寫預設值，請SchedulableMemory特別為您的叢集組態定義 的自訂值。

若要檢查運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。此命令會傳回節點的硬體組態，包括 [RealMemory](#)值。如需詳細資訊，請參閱[slurmd -C](#)。

確定運算節點的作業系統程序有足夠的記憶體。若要這樣做，請將 SchedulableMemory值設定為低於slurmd `-C`命令傳回RealMemory的值，以限制任務可用的記憶體。

使用 Slurm 進行多個執行個體類型配置

從 3.3.0 AWS ParallelCluster 版開始，您可以設定叢集從運算資源的一組已定義的執行個體類型進行配置。配置可以根據 Amazon EC2 機群低成本或最佳容量策略。

這組已定義的執行個體類型必須全部具有相同數量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相同加速器數量。如果 [Efa / Enabled](#) 設定為 true，則執行個體必須支援 EFA。如需詳細資訊和需求，請參閱 [Scheduling // SlurmQueues AllocationStrategy](#)和 [ComputeResources / Instances](#)。

您可以capacity-optimized根據您的 [CapacityType](#) 組態[AllocationStrategy](#)，將 設定為 lowest-price或。

在 [中Instances](#)，您可以設定一組執行個體類型。

Note

從 3.7.0 AWS ParallelCluster 版開始，如果您在[執行個體](#)中設定多個執行個體類型，EnableMemoryBasedScheduling則可以啟用。
對於 3.2.0 到 3.6.x AWS ParallelCluster 版，如果您在[執行個體](#)中設定多個執行個體類型，則EnableMemoryBasedScheduling無法啟用。

下列範例示範如何查詢 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 具有 64 個核心、EFA 支援和 arm64 架構的查詢。

```
$ 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
```

下一個範例叢集組態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 支援使用 Slurm 的省 Slurm 電工具外掛程式動態擴展叢集的方法。如需詳細資訊，請參閱 Slurm 文件中的 [雲端排程指南](#) 和 [Slurm 省電指南](#)。下列主題說明每個版本 Slurm 的策略。

主題

- [Slurm 3.8.0 版中的動態節點配置策略](#)
- [Slurm 3.7.x 版中的動態節點配置策略](#)
- [Slurm 3.6.x 版和舊版中的動態節點配置策略](#)

Slurm 3.8.0 版中的動態節點配置策略

從 ParallelCluster 3.8.0 版開始，ParallelCluster 使用任務層級恢復或任務層級擴展作為預設動態節點分配策略來擴展叢集：ParallelCluster 會根據每個任務的需求、分配給任務的節點數量，以及需要恢復哪些節點來擴展叢集。ParallelCluster 會從 SLURM_RESUME_FILE 環境變數取得此資訊。

動態節點的擴展是一個兩步驟程序，涉及啟動 EC2 執行個體，以及將啟動的 Amazon EC2 執行個體指派給 Slurm 節點。這兩個步驟都可以使用 all-or-nothing 或最佳嘗試邏輯來完成。

若要啟動 Amazon EC2 執行個體：

- all-or-nothing 啟動 Amazon EC2 API，目標下限等於總目標容量
- 盡最大努力呼叫最低目標等於 1 且總目標容量等於請求容量的啟動 Amazon EC2 API

若要將 Amazon EC2 執行個體指派給 Slurm 節點：

- 只有在可以為每個請求的 Slurm 節點指派 Amazon EC2 執行個體時，all-or-nothing 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會轉換為 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 任務層級擴展 are:</caption>

all-or-nothing 擴展：

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，這需要請求的運算節點成功啟動所需的所有執行個體。這可確保叢集僅在每個任務所需的容量可用時擴展，避免在擴展程序結束時留下閒置的執行個體。

策略使用 all-or-nothing 邏輯來啟動每個任務的 Amazon EC2 執行個體，以及將 Amazon EC2 執行個體指派給 Slurm 節點的 all-or-nothing 邏輯。

策略群組會分批啟動請求，每個請求的運算資源各一個，每個節點最多 500 個。對於跨越多個運算資源或超過 500 個節點的請求，ParallelCluster 會依序處理多個批次。

任何單一資源批次的失敗都會導致所有相關未使用容量的終止，確保擴展程序結束時不會留下閒置的執行個體。

限制

- 擴展所需的時間與每次執行 Slurm 繼續程式時提交的任務數量直接成正比。
- 擴展操作受限於 RunInstances 資源帳戶限制，預設為 1000 個執行個體。此限制符合 AWS EC2 API 限流政策，如需詳細資訊，請參閱 [Amazon EC2 API 限流文件](#)
- 當您在具有單一執行個體類型的運算資源中提交任務時，在跨越多個可用區域的佇列中，只有在單一可用區域中可以提供所有容量時，all-or-nothing EC2 啟動 API 呼叫才會成功。
- 當您在具有多個執行個體類型的運算資源中提交任務時，在具有單一可用區域的佇列中，只有在所有容量都可以由單一執行個體類型提供時，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，在跨越多個可用區域的佇列中，不支援 all-or-nothing Amazon EC2 啟動 API 呼叫，而 ParallelCluster 會改為執行最佳嘗試擴展。

greedy-all-or-nothing 擴展：

此all-or-nothing策略變體仍可確保叢集僅在每個任務所需的容量可用時擴展，避免擴展程序結束時的閒置執行個體，但涉及 ParallelCluster 啟動 Amazon EC2 啟動執行個體 API 呼叫，其目標容量下限為 1，嘗試將啟動的節點數量最大化至請求的容量。此策略使用最佳努力邏輯來啟動所有任務的 EC2 執行個體，以及將 Amazon EC2 執行個體指派給每個任務Slurm節點all-or-nothing邏輯。

策略群組會分批啟動請求，每個請求的運算資源各一個，每個節點最多 500 個。對於跨越多個運算資源或超過 500 個節點的請求，ParallelCluster 會依序處理多個批次。

它可確保擴展程序結束時不會留下閒置的執行個體，方法是在擴展程序期間以暫時過度擴展的成本將輸送量最大化。

限制

- 暫時過度擴展是可能的，導致在擴展完成之前轉換為執行中狀態的執行個體產生額外的成本。
- 根據 AWS RunInstances 資源帳戶限制，適用all-or-nothing策略相同的執行個體限制。

最佳嘗試擴展：

此策略會以 1 的最小容量為目標來呼叫 Amazon EC2 啟動執行個體 API 呼叫，如果並非所有請求的容量都可用，則以擴展程序執行後離開閒置執行個體的成本達到請求的總容量。此策略使用最佳努力邏輯來啟動所有任務的 Amazon EC2 執行個體，以及為每個任務指派 Amazon EC2 執行個體至 Slurm 節點的最佳努力邏輯。

策略群組會分批啟動請求，每個請求的運算資源各一個，每個節點最多 500 個。對於跨越多個運算資源或超過 500 個節點的請求，ParallelCluster 會依序處理多個批次。

此策略允許擴展遠超過多個擴展程序執行的預設 1000 個執行個體限制，而成本是在不同擴展程序中擁有閒置執行個體。

限制

- 擴展程序結束時可能的閒置執行中執行個體，適用於無法配置任務請求的所有節點的情況。

以下範例顯示動態節點的擴展如何使用不同的 ParallelCluster 啟動策略。假設您已提交兩個任務，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擴展：

- 對於第一個任務，呼叫all-or-nothing Amazon 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
- 成功將 20 all-or-nothing 啟動的執行個體指派給第一個任務的Slurm節點
- 呼叫另一個all-or-nothing Amazon EC2 啟動執行個體 API，為第二個任務請求 20 個執行個體。呼叫不成功，因為只有其他 10 個執行個體的容量。目前未啟動任何執行個體

greedy-all-or-nothing 擴展：

- 呼叫盡最大努力的 Amazon EC2 啟動執行個體 API，請求 40 個執行個體，這是所有任務請求的總容量。這會導致啟動 30 個執行個體
- 成功指派 20 all-or-nothing 啟動的執行個體至第一個任務的Slurm節點
- 嘗試將其餘啟動的執行個體指派給第二個任務Slurm節點的另一個all-or-nothing指派，但由於任務請求總數 20 中只有 10 個可用執行個體，因此指派不成功
- 10 個未指派的啟動執行個體已終止

最佳嘗試擴展：

- 呼叫盡最大努力的 Amazon EC2 啟動執行個體 API，請求 40 個執行個體，這是所有任務請求的總容量。這會導致啟動 30 個執行個體。
- 成功將 20 個啟動的執行個體指派到第一個任務的Slurm節點。
- 即使請求的總容量為 20，其餘 10 個啟動的執行個體也會成功指派給第二個任務的Slurm節點。但是，由於任務正在請求 20 個節點，並且可以將 Amazon EC2 執行個體指派給其中只有 10 個節點，因此任務無法啟動，且執行個體會保持閒置狀態，直到找到足夠的容量在稍後呼叫擴展程序時啟動缺少的 10 個執行個體，或排程器在其他已執行的運算節點上排程任務為止。

Slurm 3.7.x 版中的動態節點配置策略

ParallelCluster 使用 2 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進行配置：
 - 所有節點繼續或節點清單擴展：

ParallelCluster 只會在 Slurm的ResumeProgram執行時，根據 請求Slurm的節點清單名稱來擴展叢集。它只會依節點名稱將運算資源配置給節點。節點名稱清單可以跨越多個任務。

- 任務層級恢復或任務層級擴展：

ParallelCluster 會根據每個任務的需求、目前分配給任務的節點數量，以及需要繼續的節點，來擴展叢集。ParallelCluster 會從SLURM_RESUME_FILE環境變數取得此資訊。

- 使用 Amazon EC2 啟動策略進行配置：

- 最佳嘗試擴展：

ParallelCluster 使用目標容量下限等於 1 的 Amazon EC2 啟動執行個體 API 呼叫來擴展叢集，以啟動部分但不一定是支援請求節點所需的所有執行個體。

- All-or-nothing擴展：

ParallelCluster 使用 Amazon EC2 啟動執行個體 API 呼叫來擴展叢集，只有在啟動支援請求節點所需的所有執行個體時，該呼叫才會成功。在此情況下，它會呼叫目標容量下限等於總請求容量的 Amazon EC2 啟動執行個體 API。

根據預設，ParallelCluster 使用節點清單擴展搭配最佳 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，以便為提交的工作負載提供服務。

從 ParallelCluster 3.7.0 版開始，ParallelCluster 使用任務層級擴展搭配all-or-nothing EC2 啟動策略，用於以獨家模式提交的任務。當您以獨佔模式提交任務時，該任務具有其配置節點的獨佔存取權。如需詳細資訊，請參閱 Slurm 文件中的 [EXCLUSIVE](#)。

若要以排他性模式提交任務：

- 將Slurm任務提交至叢集時傳遞獨佔旗標。例如 `sbatch ... --exclusive`。

或

- 將任務提交至已設定為 [JobExclusiveAllocation](#) 的叢集佇列true。

在排他性模式下提交任務時：

- ParallelCluster 目前會批次處理啟動請求，以包含最多 500 個節點。如果任務請求超過 500 個節點，則 ParallelCluster 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。
- 如果節點配置位於單一運算資源中，則 ParallelCluster 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。如果啟動請求失敗，ParallelCluster 會終止所有啟動請求建立的未使用容量。

- 如果節點配置跨越多個運算資源，則 ParallelCluster 需要為每個運算資源提出all-or-nothing啟動請求。這些請求也會批次處理。如果其中一個運算資源的啟動請求失敗，ParallelCluster 會終止所有運算資源啟動請求建立的未使用容量。

具有all-or-nothing啟動策略已知限制的任務層級擴展：

- 當您在具有單一執行個體類型的運算資源中提交任務時，在跨越多個可用區域的佇列中，只有在單一可用區域中可提供所有容量時，all-or-nothing EC2 啟動 API 呼叫才會成功。
- 當您在具有多個執行個體類型的運算資源中提交任務時，在具有單一可用區域的佇列中，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，在跨越多個可用區域的佇列中，不支援all-or-nothingAmazon EC2 啟動 API 呼叫，而 ParallelCluster 會改為執行最佳嘗試擴展。

Slurm 3.6.x 版和舊版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進行配置：
 - 所有節點恢復或節點清單擴展：ParallelCluster 只會在 Slurm的ResumeProgram 執行時，根據 Slurm請求的節點清單名稱擴展叢集。它只會依節點名稱將運算資源配置給節點。節點名稱清單可以跨越多個任務。
- 使用 Amazon EC2 啟動策略進行配置：
 - 最佳努力擴展：ParallelCluster 使用目標容量下限等於 1 的 Amazon EC2 啟動執行個體 API 呼叫來擴展叢集，以啟動部分但不一定是支援請求節點所需的所有執行個體。

ParallelCluster 使用節點清單擴展搭配最佳 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，以便為提交的工作負載提供服務。

限制

- 擴展程序結束時可能的閒置執行中執行個體，適用於無法配置任務請求的所有節點的情況。

Slurm 使用 會計 AWS ParallelCluster

從 3.3.0 版開始，AWS ParallelCluster 支援使用叢集組態參數 [SlurmSettings/資料庫](#) 進行Slurm會計。

從 3.10.0 版開始，AWS ParallelCluster 支援使用叢集組態參數 [SlurmSettings](#) / [ExternalSlurmdbd](#) 的外部 Slurmdbd [ExternalSlurmdbd](#) 進行 Slurm 會計。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來執行下列動作：

- 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 Slurm 更進階的功能，例如資源限制強制執行、公平共用和 QOSs。
- 收集並儲存任務資料，例如執行任務的使用者、任務的持續時間及其使用的資源。您可以使用 `sacct` 公用程式檢視儲存的資料。

Note

AWS ParallelCluster 支援 Slurm 計算 [Slurm 支援的 MySQL 資料庫伺服器](#)。

在 AWS ParallelCluster v3.10.0 和更新版本 Slurmdbd 中使用外部 進行 Slurm 會計

設定 Slurm 會計之前，您必須擁有現有的外部 Slurmdbd 資料庫伺服器，該伺服器會連接到現有的外部資料庫伺服器。

若要設定此項，請定義下列項目：

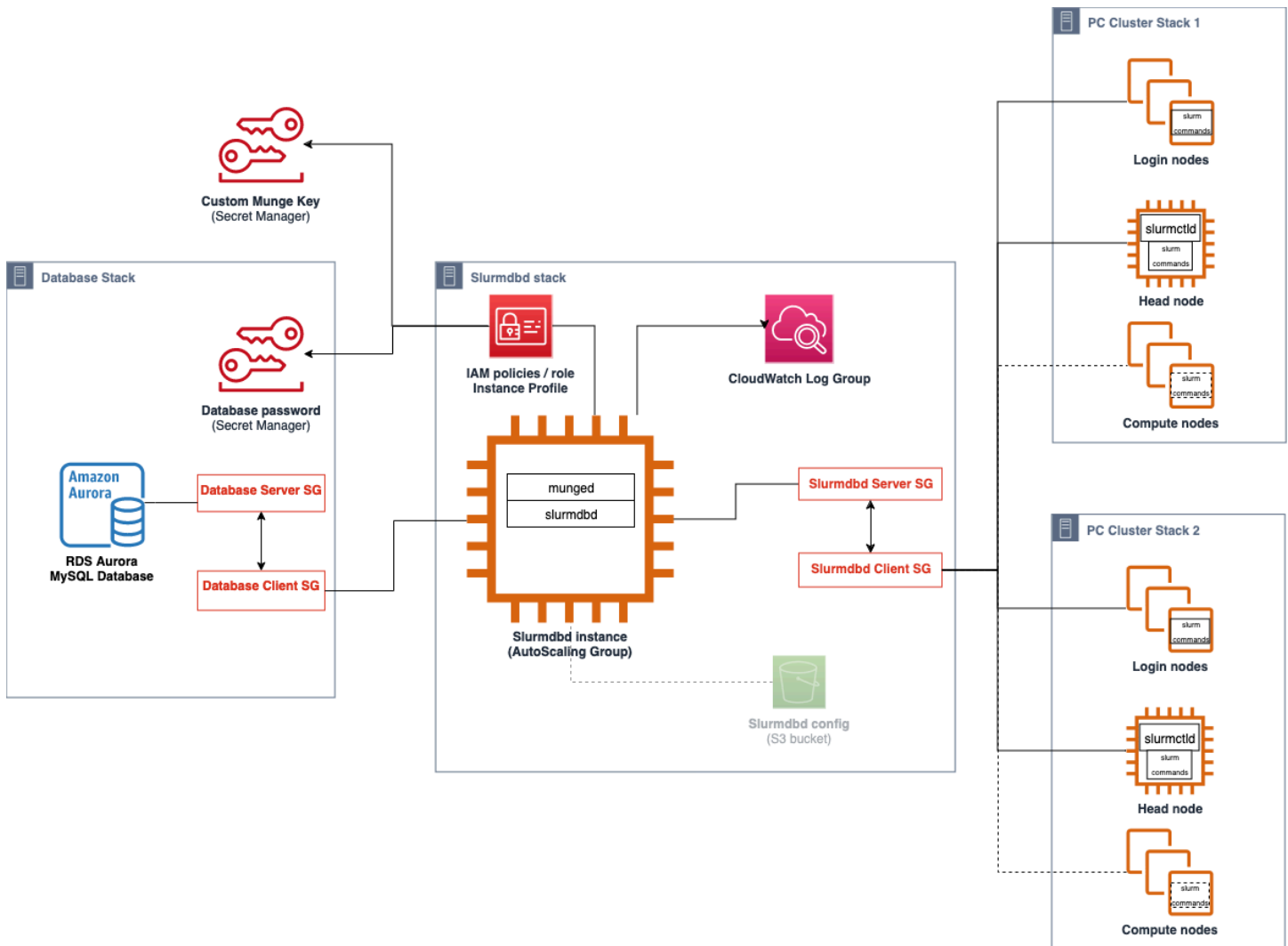
- [ExternalSlurmdbd/主機](#) 中外部 Slurmdbd 伺服器的地址。伺服器必須存在且可從前端節點連線。
- 在 [MungeKeySecretArn](#) 中與外部 Slurmdbd 伺服器通訊的 `munge` 金鑰。

若要逐步完成教學課程，請參閱 [使用外部 Slurmdbd 會計建立叢集](#)。

Note

您負責管理 Slurm 資料庫會計實體。

AWS ParallelCluster 外部 SlurmDB 支援功能的架構可啟用多個共用相同 SlurmDB 和相同資料庫的叢集。



⚠ Warning

AWS ParallelCluster 與外部 之間的流量SlurmDB不會加密。建議在信任的網路SlurmDB中執行叢集和外部。

在 AWS ParallelCluster v3.3.0 和更新Slurmdbd版本中使用前端節點進行Slurm會計

設定Slurm會計之前，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 設定Slurm會計 AWS ParallelCluster，您必須定義下列項目：

- [資料庫/Uri](#) 中外部資料庫伺服器的 URI。伺服器必須存在且可從前端節點連線。

- 存取 [資料庫 / PasswordSecretArn](#) 和 [資料庫 / UserName](#) 中定義的外部資料庫的登入資料。AWS ParallelCluster 會使用此資訊在 Slurm 層級設定會計作業，並在前端節點上設定 slurmdbd 服務。slurmdbd 是管理叢集與資料庫伺服器之間通訊的協助程式。

若要逐步完成教學課程，請參閱 [使用Slurm會計建立叢集](#)。

Note

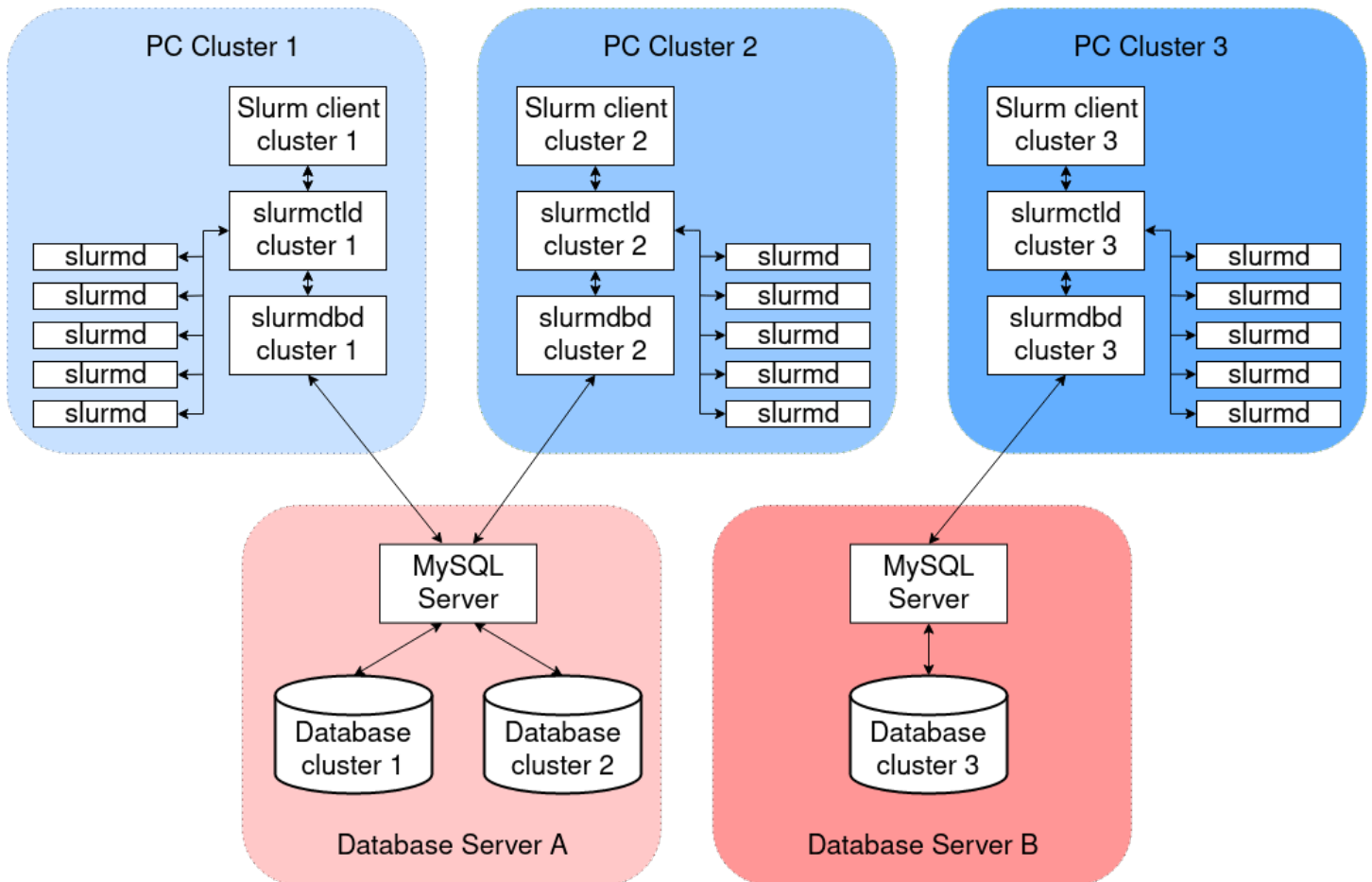
AWS ParallelCluster 透過將預設叢集使用者設定為 Slurm 資料庫中的資料庫管理員，執行 Slurm 會計資料庫的基本引導。AWS ParallelCluster 不會將任何其他使用者新增至會計資料庫。客戶負責管理 Slurm 資料庫中的會計實體。

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。相同的資料庫伺服器可以跨多個叢集使用，但每個叢集都有自己的個別資料庫。AWS ParallelCluster 會使用叢集名稱在 slurmdbd 組態檔案 [StorageLoc](#) 參數中定義資料庫的名稱。請考慮下列情況。資料庫伺服器上存在的資料庫包含未對應至作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的叢集，以對應至該資料庫。會為新叢集 Slurm 重複使用資料庫。

Warning

- 我們不建議同時設定多個叢集來使用相同的資料庫。這樣做可能會導致效能問題，甚至是資料庫死結情況。
- 如果在叢集的前端節點上啟用 Slurm 會計，我們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 會計可能會對叢集的前端節點增加壓力。

在會計功能的目前架構中 AWS ParallelCluster Slurm，每個叢集都有自己的 slurmdbd 協助程式執行個體，如下圖範例組態所示。



如果您要將自訂Slurm多叢集或聯合功能新增至叢集環境，則所有叢集都必須參考相同的slurmdbd執行個體。對於此替代方案，我們建議您在一個叢集上啟用 AWS ParallelCluster Slurm會計，並手動設定其他叢集以連線到在第一個叢集上託管slurmdbd的。

如果您使用的 AWS ParallelCluster 是 3.3.0 版之前的版本，請參閱本 [HPC 部落格文章](#) 中所述的實作 Slurm會計的替代方法。

Slurm 會計考量

不同 VPCs 上的資料庫和叢集

若要啟用Slurm會計，資料庫伺服器需要做為slurmdbd協助程式執行的讀取和寫入操作的後端。在建立或更新叢集以啟用Slurm會計之前，前端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：

- 若要在叢集端slurmdbd的與資料庫伺服器之間啟用通訊，您必須在兩個 VPCs 之間設定連線。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](#)。

- 您必須建立要連接到叢集 VPC 上前端節點的安全群組。在兩個 VPCs 對等互連之後，即可在資料庫端和叢集端安全群組之間進行交叉連結。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](#)。

在 `slurmdbd` 和資料庫伺服器之間設定 TLS 加密

使用 AWS ParallelCluster 提供的預設 Slurm 會計組態，如果伺服器支援 TLS encryption. AWS database 服務，例如 Amazon RDS，且預設 Amazon Aurora 支援 TLS 加密，則會 `slurmdbd` 建立與資料庫伺服器的 TLS 加密連線。

您可以在資料庫伺服器上設定 `require_secure_transport` 參數，在伺服器端要求安全連線。這是在提供的 CloudFormation 範本中設定。

遵循安全最佳實務，我們建議您也在 `slurmdbd` 用戶端上啟用伺服器身分驗證。若要這樣做，請在中設定 [StorageParameters](#) `slurmdbd.conf`。將伺服器 CA 憑證上傳至叢集的前端節點。接著，將 `StorageParameters` 中的 [SSL_CA](#) 選項 `slurmdbd.conf` 設定為前端節點上伺服器 CA 憑證的路徑。這樣做可在 `slurmdbd` 端啟用伺服器身分驗證。進行這些變更後，請重新啟動 `slurmdbd` 服務，在啟用身分驗證的情況下重新建立與資料庫伺服器的連線。

更新資料庫登入資料

若要更新 [資料庫/UserName](#) 或 [PasswordSecretArn](#) 的值，您必須先停止運算機群。假設存放在秘密中的 AWS Secrets Manager 秘密值已變更，且其 ARN 不會變更。在這種情況下，叢集不會自動將資料庫密碼更新為新值。若要更新叢集的新秘密值，請從前端節點執行下列命令。

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

Warning

為了避免遺失會計資料，建議您只在運算機群停止時變更資料庫密碼。

資料庫監控

建議您啟用 AWS 資料庫服務的監控功能。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

從 3.6.0 AWS ParallelCluster 版開始，您可以在 AWS ParallelCluster 叢集 `slurm.conf` Slurm 組態中自訂組態。

在叢集組態中，您可以使用下列叢集組態設定來自訂組態 Slurm 參數：

- 如果您同時指定 Slurm / [SlurmSettings](#) [CustomSlurmSettings](#) 或 `parameter. AWS ParallelCluster fails`，即可自訂整個叢集的 [CustomSlurmSettingsIncludeFile](#) 參數。
- 使用 [SlurmQueues](#) / [CustomSlurmSettings](#) (映射到 Slurm 分割區) 自訂佇列的 Slurm 參數。
- 使用 [SlurmQueues](#) / [CustomSlurmSettings](#) (映射至 Slurm 節點) [ComputeResources](#) 自訂運算資源的 Slurm 參數。

Slurm 使用時的組態自訂限制和考量事項 AWS ParallelCluster

- 對於 `CustomSlurmSettings` 和 `CustomSlurmSettingsIncludeFile` 設定，您只能指定和更新包含在您用來設定叢集的 [Slurm 版本](#) 所支援的 AWS ParallelCluster 版本中的 `slurm.conf` 參數。
- 如果您在任何 `CustomSlurmSettings` 參數中指定自訂 Slurm 組態，會 AWS ParallelCluster 執行驗證檢查，並防止設定或更新與 AWS ParallelCluster 邏輯衝突的 Slurm 組態參數。已知與衝突的 Slurm 組態參數 AWS ParallelCluster 會在拒絕清單中識別。如果新增其他 Slurm 功能，拒絕清單可能會在未來 AWS ParallelCluster 版本中變更。如需詳細資訊，請參閱 [拒絕列出的 Slurm 組態參數 CustomSlurmSettings](#)。
- AWS ParallelCluster 只會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。您有責任驗證您的自訂 Slurm 組態參數。無效的自訂 Slurm 組態參數可能會導致 Slurm 協助程式失敗，進而導致叢集建立和更新失敗。
- 如果您在 `中` 指定自訂 Slurm 組態 `CustomSlurmSettingsIncludeFile`，AWS ParallelCluster 不會執行任何驗證。
- 您可以更新 `CustomSlurmSettings` 和 `CustomSlurmSettingsIncludeFile` 而無需停止和啟動運算機群。在此情況下，`slurmctld` 會 AWS ParallelCluster 重新啟動協助程式並執行 `scontrol reconfigure` 命令。

在整個叢集中註冊變更之前，某些 Slurm 組態參數可能需要不同的操作。例如，它們可能需要重新啟動叢集中的所有協助程式。您有責任驗證 AWS ParallelCluster 操作是否足以在更新期間傳播您的自訂 Slurm 組態參數設定。如果您發現 AWS ParallelCluster 操作不足，您有責任提供傳播更新設定所需的其他動作，如 [Slurm 文件](#) 中所建議。

拒絕列出的Slurm組態參數 **CustomSlurmSettings**

下表列出拒絕使用的 參數 AWS ParallelCluster 版本，從 3.6.0 版開始。CustomSlurmSettings不支援 3.6.0 版之前的 AWS ParallelCluster 版本。

叢集層級的拒絕列出參數：

Slurm 參數	AWS ParallelCluster 版本中列入拒絕清單
CommunicationParameters	3.6.0
Epilog	3.6.0
GresTypes	3.6.0
LaunchParameters	3.6.0
Prolog	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	AWS ParallelCluster 版本中列入拒絕清單
TaskPlugin	3.6.0
TreeWidth	3.6.0

在叢集組態中設定[原生Slurm會計整合](#)時，叢集層級的拒絕列出參數：

Slurm 參數	AWS ParallelCluster 版本中列入拒絕清單
AccountingStorageType	3.6.0
AccountingStorageHost	3.6.0
AccountingStoragePort	3.6.0
AccountingStorageUser	3.6.0
JobAcctGatherType	3.6.0

佇列管理的佇列在佇列（分割區）層級拒絕列出的參數 AWS ParallelCluster：

Slurm 參數	AWS ParallelCluster 版本中列入拒絕清單
節點	3.6.0
PartitionName	3.6.0
ResumeTimeout	3.6.0
State	3.6.0
SuspendTime	3.6.0

運算資源的運算資源（節點）層級拒絕列出的參數，管理者為 AWS ParallelCluster：

Slurm 參數	拒絕列出 AWS ParallelCluster 版本 和更新版本
CPUs	3.6.0

Slurm 參數	拒絕列出 AWS ParallelCluster 版本 和更新版本
功能	3.6.0
Gres	3.6.0
NodeAddr	3.6.0
NodeHostname	3.6.0
NodeName	3.6.0
Weight	3.7.0

Slurm 與 **prologepilog**

從 3.6.0 AWS ParallelCluster 版開始，使用 AWS ParallelCluster 部署的 Slurm 組態包含 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
```

如需詳細資訊，請參閱 Slurm 文件中的 [Prolog 和 Epilog 指南](#)。

AWS ParallelCluster 包含下列 prolog 和 epilog 指令碼：

- 90_plcluster_health_check_manager (在 Prolog 資料夾中)
- 90_pcluster_noop (在 Epilog 資料夾中)

Note

Prolog 和 Epilog 資料夾都必須包含至少一個檔案。

您可以將自訂 prolog 或 epilog 指令碼新增至對應的 Prolog 和 Epilog 資料夾，以使用它們。

⚠ Warning

Slurm 會以反向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任務所需的時間。在執行多個或長時間執行的 prolog 指令碼時更新 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，請在個別的 Prolog 和 Epilog 資料夾中找到指令碼。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以擴展的運算節點數量所定義。運算節點由 AWS ParallelCluster 組態中運算資源內定義的 Amazon EC2 執行個體支援 (Scheduling/SlurmQueues/ [ComputeResources](#))，並組織成 1 : 1 (Scheduling/[SlurmQueues](#)) 對應至 Slurm 分割區的佇列。

在運算資源中，您可以設定必須在叢集中持續執行的運算節點（執行個體）數目下限 ([MinCount](#))，以及運算資源可擴展至 ([MaxCount3](#)) 的執行個體數目上限。

在叢集建立時間或叢集更新時，MinCount 會針對叢集中定義的每個運算資源 (Scheduling/SlurmQueues/ [ComputeResources](#))，依在中設定的數目 AWS ParallelCluster 啟動 Amazon EC2 執行個體。啟動以涵蓋叢集中運算資源最小數量節點的執行個體稱為靜態節點。啟動後，除非發生特定事件或條件，否則靜態節點應持續存在於叢集中，系統不會終止這些節點。例如，這類事件包括 Slurm 或 Amazon EC2 運作狀態檢查失敗，以及 Slurm 節點狀態變更為 DRAIN 或 DOWN。

在 1 到 'MaxCount - MinCount' (MaxCount 減去 MinCount)，隨需啟動以處理叢集負載增加的 Amazon EC2 執行個體，稱為動態節點。它們的性質是暫時性的，它們被啟動來提供待命任務，一旦它們在叢集組態 Scheduling/SlurmSettings/[ScaledownIdleTime](#) 中保持閒置一段時間後就會終止（預設值：10 分鐘）。

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，假設有下列 AWS ParallelCluster 組態：

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

下列節點將在 中定義 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]
```

當運算資源具有時 `MinCount == MaxCount`，所有對應的運算節點都會是靜態的，而且所有執行個體都會在叢集建立/更新時啟動，並保持啟動和執行。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、運算資源或變更運算資源 MinCount/MaxCount 的。從 3.9.0 AWS ParallelCluster 版開始，減少佇列的大小需要停止運算機群，或將的 [QueueUpdateStrategy](#) 設定為 TERMINATE，才能進行叢集更新。在下列情況下，不需要停止運算機群或將 [QueueUpdateStrategy](#) 設定為 TERMINATE：

- 將新佇列新增至排程/[SlurmQueues](#)
- 將新的運算資源 Scheduling/[SlurmQueues](#)/[ComputeResources](#) 新增至佇列
- 增加運算資源 [MaxCount](#) 的
- 增加運算資源的 MinCount 並增加相同運算資源至少相同數量的 MaxCount

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。

- 從名稱為 <Queue/Name>-* Scheduling/[SlurmQueues](#) 的所有運算節點移除佇列時，靜態和動態都會從 Slurm 組態中移除，且對應的 Amazon EC2 執行個體將會終止。
- Scheduling/[SlurmQueues](#)/[ComputeResources](#) 從佇列移除運算資源時，所有名為的運算節點 <Queue/Name>-*-<ComputeResource/Name>-*，包括靜態和動態，都會從 Slurm 組態中移除，且對應的 Amazon EC2 執行個體將會終止。

變更運算資源的 MinCount 參數時，我們可以區分兩種不同的案例，如果 MaxCount 保持等於 MinCount（僅限靜態容量），如果 MaxCount 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 MinCount == MaxCount，增加 MinCount（和）MaxCount 時，叢集的設定方式是將靜態節點數量擴展到的新值，MinCount<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount> 而且系統會繼續嘗試啟動 Amazon EC2 執行個體，以滿足新的必要靜態容量。
- 如果 MinCount == MaxCount，則在減少 MinCount（和 MaxCount）數量 N 時，將透過移除最後一個 N 個靜態節點來設定叢集，<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>] 且系統會終止對應的 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]
```

- 在 MinCount 和 -30 上更新 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 數量 N 時（假設 MaxCount 將保持不變），叢集的設定方式是將靜態節點數目擴展到 $\text{MinCount}() \text{old_MinCount} + N$ 的新值：`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` 且系統會繼續嘗試啟動 Amazon EC2 執行個體，以滿足新的必要靜態容量。此外，為了滿足運算資源的 MaxCount 容量，叢集組態會透過移除最後一個 N 個動態節點來更新：`<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` 且系統會終止對應的 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]
```

- 將 $+30$ 更新為 MinCount ： $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

```
$ 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 $\text{MinCount} < \text{MaxCount}$ ，增加 MinCount 且數量 MaxCount 相同 N 時，叢集的設定方式是將靜態節點數量擴展到 $\text{MinCount}() \text{ old_MinCount} + N$ 的新值：， $\langle \text{Queue/Name} \rangle\text{-st-}\langle \text{ComputeResource/Name} \rangle\text{-}\langle \text{old_MinCount} + N \rangle$ 而且系統會繼續嘗試啟動 Amazon EC2 執行個體，以滿足新的必要靜態容量。此外，不會對動態節點數量進行任何變更，以遵守新的

MaxCount 值。

- 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]

```

- 將 +30 更新為 MinCount ： $\text{MinCount} = 130$ ($\text{MaxCount} = 180$)

```

$ 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 $\text{MinCount} < \text{MaxCount}$ ，則在減少 MinCount 數量 N 時（假設 MaxCount 將保持不變），則會透過移除最後一個 N 個靜態節點靜態節點來設定叢集， $\langle \text{Queue/Name} \rangle\text{-st-}\langle \text{ComputeResource/Name} \rangle\text{-}\langle \text{old_MinCount} - N \rangle \dots \langle \text{old_MinCount} \rangle$ 且系統會終止對應的 Amazon EC2 執行個體。此外，為了滿足運算資源的 MaxCount 容量，叢集組態的更新方式是擴展動態節點的數量以填補間隙 $\text{MaxCount} - \text{new_MinCount}$ ： $\langle \text{Queue/Name} \rangle\text{-dy-}\langle \text{ComputeResource/Name} \rangle\text{-}\langle 1 \dots \text{MaxCount} - \text{new_MinCount} \rangle$ 。在此情況下，由於這些是動態節點，除非排程器在新節點上有待定任務，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]

```

- 在上更新 -30 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 $\text{MinCount} < \text{MaxCount}$ ，當減少 MinCount 且數量 MaxCount 相同 N 時，將透過移除最後一個 N 個靜態節點來設定叢集，`<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] 且系統會終止對應的 Amazon EC2 執行個體。

此外，不會對動態節點數量進行任何變更，以遵守新 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]

```

- 在上更新 -30 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 $\text{MinCount} < \text{MaxCount}$ ，則在減少 MaxCount 數量 N 時（假設 MinCount 將保持不變），叢集將透過移除最後一個 N 個動態節點進行設定，`<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>]` 如果靜態節點受到 `running.No` 影響，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]
```

- 在上更新 -30 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]
```

對任務的影響

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需求，否則在移除的節點上執行的 `sbatch` 任務會重新排入佇列。在此情況下，任務會失敗，狀態為 `NODE_FAIL`，並從佇列中消失，且必須手動重新提交。

如果您打算執行叢集調整大小更新，您可以防止任務在計劃更新期間將移除的節點中執行。這可以透過設定要在維護中移除的節點來實現。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任務。

假設透過計劃的叢集調整大小更新，您將移除節點 `queue-st-computeresource-[9-10]`。您可以使用下列命令建立 Slurm 保留

```
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]
```

這將在節點 `maint_for_update` 上建立名為 `maint_for_update` 的 Slurm 保留 `queueu-st-computeresource-[9-10]`。從建立保留開始，就無法再將任何任務執行到節點 `queueu-st-computeresource-[9-10]`。請注意，保留不會阻止任務最終配置到節點 `queueu-st-computeresource-[9-10]`。

叢集調整大小更新後，如果僅在調整大小更新期間移除的節點上設定 Slurm 保留，系統會自動刪除維護保留。如果您改為在叢集調整大小更新後仍存在的節點上建立 Slurm 保留，我們可能會想要在執行調整大小更新後，使用以下命令來移除節點上的維護保留

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

如需 Slurm 保留的其他詳細資訊，請參閱 [此處](#) 的官方 SchedMD 文件。

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 AWS ParallelCluster 組態產生更新的 Slurm 分割區組態
- 重新啟動 `slurmctld` (透過 Chef 服務配方完成)
- 檢查 `slurmctld` 狀態 (`systemctl is-active --quiet slurmctld.service`)
- 重新載入 Slurm 組態 (`scontrol reconfigure`)
- 啟動 `clustermgtd` (`supervisorctl start clustermgtd`)

搭配使用 AWS Batch (`awsbatch`) 排程器 AWS ParallelCluster

Warning

AWS CodeBuild 亞太區域 (馬來西亞) (`ap-southeast-5`) 和亞太區域 (泰國) (`ap-southeast-7`) 區域不支援。因此，這些區域不支援 ParallelCluster AWS Batch 整合。

AWS ParallelCluster 也支援 AWS Batch 排程器。下列主題說明如何使用 AWS Batch。如需的詳細資訊 AWS Batch，請參閱 [AWS Batch](#)。如需文件，請參閱 [AWS Batch 使用者指南](#)。

AWS ParallelCluster 的 CLI 命令 AWS Batch

當您使用 `awsbatch` 排程器時，的 AWS ParallelCluster AWS Batch CLI 命令會自動安裝在 AWS ParallelCluster 前端節點中。CLI 使用 AWS Batch API 操作並允許下列操作：

- 提交和管理任務。
- 監控任務、佇列和主機。
- 鏡像傳統排程器命令。

Important

AWS ParallelCluster 不支援的 GPU 任務 AWS Batch。如需詳細資訊，請參閱 [GPU 任務](#)。

此 CLI 會以個別的套件形式分佈。如需詳細資訊，請參閱 [排程器支援](#)。

主題

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)
- [awsbqueues](#)
- [awsbhosts](#)

awsbsub

將任務提交至叢集的任務佇列。

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


⚠ Important

AWS ParallelCluster 不支援的 GPU 任務 AWS Batch。如需詳細資訊，請參閱 [GPU 任務](#)。

定位引數**command**

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

arguments

(選用) 指定命令或命令檔案的引數。

具名引數**-jn *JOB_NAME*, --job-name *JOB_NAME***

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

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

指定要使用的叢集。

-cf, --command-file

指出命令是要傳輸至運算執行個體的檔案。

預設：False

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

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

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

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

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

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

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

指定要保留給容器的 vCPU 數目。與 `-nodes` 搭配使用時，它會識別每個節點 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)。如果您的任務嘗試超過此處指定的記憶體限制，則任務會結束。

預設：128

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

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

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

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

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

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

預設：1

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

以秒為單位指定持續時間（從任務嘗試的 `startedAt` 時間戳記測量），之後如果任務尚未完成，則會 AWS Batch 終止任務。逾時值必須至少為 60 秒。

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

指定要為任務保留的節點數目。指定此參數的值，以啟用多節點平行提交。

Note

當 [Scheduler](#) / [AwsBatchQueues](#) / [CapacityType](#) 參數設定為 時 SPOT，不支援多節點平行任務。此外，您的帳戶中必須有 `AWSServiceRoleForEC2Spot` 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 Elastic Compute Cloud 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，而無需指定陣列任務的任務 ID。序列相依性允許每個子陣列任務循序完成，從索引 0 開始。您也可以指定 N_TO_N 類型相依性，以及陣列任務的任務 ID。N_TO_N 相依性表示，此任務的每個索引子系必須等待各相依性對應的索引子系完成後，才能開始。此參數的語法為 "jobId=<*string*>, type=<*string*>; ..."。

awsbstat

顯示在叢集的任務佇列中提交的任務。

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

定位引數***job_ids***

指定以空格分隔的清單，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 ... ]]
```

定位引數

job_queues

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

具名引數

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

指定要使用的叢集名稱。

-d, --details

指出是否顯示佇列的詳細資訊。

預設：False

awsbhosts

顯示屬於叢集運算環境的主機。

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

定位引數

instance_ids

指定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 for ONTAP](#) 和 [FSx for OpenZFS](#) 共用儲存磁碟區、[Amazon EFS](#) 和 [FSx for Lustre](#) 共用儲存檔案系統，或[檔案快取](#)。建議您遵循[AWS 架構良好的架構可靠性支柱](#)指引，並備份您的磁碟區和檔案系統。

選取符合您 HPC 應用程式 I/O 需求的儲存系統。您可以根據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 磁碟區會連接到前端節點，並透過 NFS 與運算節點共用。此選項可以符合成本效益，但效能取決於儲存需求擴展時的前端節點資源。當更多運算節點新增至叢集且輸送量需求增加時，這可能會成為瓶頸。

Amazon EFS 檔案系統會隨著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 檔案系統在您的叢集上執行平行處理和延遲敏感的應用程式。

FSx for Lustre 檔案系統可以處理高達每秒數百 GB 輸送量、數百萬 IOPS 和低於一毫秒延遲的大量資料集。將 FSx for Lustre 檔案系統用於要求嚴苛的高效能運算環境。

在中[SharedStorage 區段](#)，您可以定義外部或 AWS ParallelCluster 受管儲存：

- 外部儲存體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體。
- 受管儲存是指 AWS ParallelCluster 已建立且可刪除的磁碟區或檔案系統。

外部儲存

您可以設定在建立或更新叢集時，將外部儲存 AWS ParallelCluster 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從叢集分離外部儲存體。您的資料會保留，而且您可以在叢集生命週期外將其用於長期永久共用儲存。

Note

3.8 AWS ParallelCluster 之前的版本不允許將外部受管檔案系統掛載在 /home。從 3.8 版開始，AWS ParallelCluster 可讓您使用 /home 做為外部受管檔案系統的掛載點。您可以透過在下將 /home 指定 /home 為 [MountDir](#) 參數的值，將外部受管檔案系統掛載至 [SharedStorage 區段](#)。

Amazon File Cache 不適合用作系統 /home 目錄，因此目前不支援掛載 /home。

在 [SharedStorage 區段 SharedStorageType](#) 組態選項下指定 /home 目錄時，將會覆寫 [SharedStorage 區段](#) 中的設定。

將外部檔案系統掛載到 /home 目錄時，會將前端節點 /home 的內容 AWS ParallelCluster 複製到外部檔案系統，而不會覆寫外部儲存體上的現有檔案。這包括在外部檔案系統上不存在時，轉移預設使用者的叢集 SSH 金鑰。如需詳細資訊，請參閱 [AWS ParallelCluster 共用儲存考量](#)。

AWS ParallelCluster 受管儲存

AWS ParallelCluster 根據預設，受管儲存取決於組態中叢集的生命週期。SharedStorage DeletionPolicy 組態參數 Delete 預設為。

根據預設，如果下列其中一項為 true，則會刪除 AWS ParallelCluster 受管檔案系統或磁碟區及其資料。

- 您刪除叢集。
- 您可以變更受管共用儲存組態 Name。
- 您可以從組態中移除受管共用儲存。

DeletionPolicy 設定為 Retain 以保留您的受管共用檔案系統或磁碟區和資料。建議您定期備份資料，以避免資料遺失。您可以使用 [AWS Backup](#) 集中管理所有儲存選項的備份。

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 受管儲存體轉換為外部儲存體](#)。

如需共用儲存配額的資訊，請參閱 [共用儲存的配額](#)。

如需共用儲存和切換到新 AWS ParallelCluster 版本的詳細資訊，請參閱 [最佳實務：將叢集移至新的 AWS ParallelCluster 次要或修補程式版本](#)。

您可以設定在建立或更新叢集時，將外部儲存 AWS ParallelCluster 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從叢集分離外部儲存體。您的資料會保留，而且您可以將其用於與叢集生命週期無關的長期永久共用儲存解決方案。

根據預設，受管儲存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 受管儲存體轉換為外部儲存體](#)。

透過特定設定，您可以針對使用案例最佳化每個支援的儲存解決方案。

如需共用儲存配額，請參閱 [共用儲存的配額](#)。

如需共用儲存和切換到新 AWS ParallelCluster 版本的詳細資訊，請參閱 [最佳實務：將叢集移至新的 AWS ParallelCluster 次要或修補程式版本](#)。

下列主題說明如何為 AWS ParallelCluster 支援的每個儲存服務設定共用儲存。

主題

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx for Lustre](#)
- [設定 FSx for ONTAP、FSx for OpenZFS 和檔案快取共用儲存](#)
- [在 中使用共用儲存 AWS ParallelCluster](#)
- [共用儲存的配額](#)

Amazon Elastic Block Store

若要將現有的外部 Amazon EBS 磁碟區用於與叢集生命週期無關的長期永久儲存，請指定 [EbsSettings](#) / [VolumeId](#)。

如果您未指定 [VolumeId](#)，根據預設，AWS ParallelCluster 會在建立叢集 [EbsSettings](#) 時從 建立受管 EBS 磁碟區。當叢集刪除或從叢集組態中移除磁碟區時，AWS ParallelCluster 也會刪除磁碟區和資料。

對於 AWS ParallelCluster 受管 EBS 磁碟區，您可以在刪除叢集或從叢集組態中移除 Snapshot 磁碟區時 Retain，使用 [EbsSettings](#) / [DeletionPolicy](#) AWS ParallelCluster 來指示 Delete、或 磁碟區。DeletionPolicy 預設會設定為 Delete。

⚠ Warning

對於 AWS ParallelCluster 受管共用儲存，DeletionPolicy Delete 預設為。這表示，如果下列其中一項為 true，則會刪除受管磁碟區及其資料：

- 您刪除叢集。
- 您可以變更受管共用儲存組態 [SharedStorage / Name](#)。
- 您可以從組態中移除受管共用儲存。

建議您定期使用快照備份共用儲存體，以避免資料遺失。如需 Amazon EBS 快照的詳細資訊，請參閱《Amazon Elastic Compute Cloud Linux [執行個體使用者指南](#)》中的 [Amazon EBS 快照](#)。若要了解如何管理跨的資料備份 AWS 服務，請參閱《AWS Backup 開發人員指南》中的 [AWS 備份](#)。

Amazon Elastic File System

若要在叢集生命週期外使用現有的外部 Amazon EFS 檔案系統進行長期永久儲存，請在建立叢集 [EfsSettings](#) 時指定 [EfsSettings / FileSystemId](#)，預設情況下，會從 AWS ParallelCluster 建立受管 Amazon EFS 檔案系統。當叢集刪除或從叢集組態中移除檔案系統時，AWS ParallelCluster 也會刪除檔案系統和資料。

對於 AWS ParallelCluster 受管 Amazon EFS 檔案系統，您可以使用 [EfsSettings / DeletionPolicy](#) AWS ParallelCluster 來指示 Delete，Retain 或在刪除叢集時，或從叢集組態中移除檔案系統時指示 或。DeletionPolicy 預設會設定為 Delete。

⚠ Warning

對於 AWS ParallelCluster 受管共用儲存，DeletionPolicy Delete 預設為。這表示，如果下列其中一項為 true，則會刪除受管檔案系統及其資料：

- 您刪除叢集。
- 您可以變更受管共用儲存組態 [SharedStorage / Name](#)。
- 您可以從組態中移除受管共用儲存。

我們建議您定期備份共用儲存體，以避免資料遺失。如需如何備份個別 Amazon EFS 磁碟區的詳細資訊，請參閱《Amazon Elastic [File System 使用者指南](#)》中的備份 [Amazon EFS](#) 檔

案系統。Amazon Elastic File System 若要了解如何管理跨 的資料備份 AWS 服務，請參閱《AWS Backup 開發人員指南》中的[AWS 備份](#)。

Amazon FSx for Lustre

若要在叢集生命週期外使用現有的外部 FSx for Lustre 檔案系統進行長期永久儲存，請指定 [FsxLustreSettings](#) / [FileSystemId](#)。

如果您未指定 [FsxLustreSettings](#) / [FileSystemId](#)，預設情況下，會在建立叢集 [FsxLustreSettings](#) 時從 AWS ParallelCluster 建立受管 FSx for Lustre 檔案系統。當叢集刪除或從叢集組態中移除檔案系統時，AWS ParallelCluster 也會刪除檔案系統和資料。

對於 AWS ParallelCluster 受管 FSx for Lustre 檔案系統，您可以在刪除叢集或從叢集組態中移除 [DeleteRetain](#) 檔案系統時 [DeletionPolicy](#)，使用 [FsxLustreSettings](#) / AWS ParallelCluster 來指示 或 檔案系統。DeletionPolicy 預設會設定為 Delete。

Warning

對於 AWS ParallelCluster 受管共用儲存，DeletionPolicy Delete 預設為。這表示，如果下列其中一項為 true，則會刪除受管檔案系統及其資料：

- 您刪除叢集。
- 您可以變更受管共用儲存組態 [SharedStorage](#) / [Name](#)。
- 您可以從組態中移除受管共用儲存。

我們建議您定期備份共用儲存體，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](#)。若要了解如何管理跨 的資料備份 AWS 服務，請參閱《AWS Backup 開發人員指南》中的[AWS 備份](#)。

設定 FSx for ONTAP、FSx for OpenZFS 和檔案快取共用儲存

對於 FSx for ONTAP、FSx for OpenZFS 和檔案快取，您可以使用 [FsxOntapSettings](#) / [VolumeId](#)、[VolumeIdFsxOpenZfsSettings](#) / 和 [FileCacheSettings](#) / [FileCacheId](#) 來指定為叢集掛載外部現有磁碟區或檔案快取。

AWS ParallelCluster FSx for ONTAP、FSx for OpenZFS 和檔案快取不支援受管共用儲存。

在 中 使用共用儲存 AWS ParallelCluster

在下列各節中，您將了解如何使用 AWS ParallelCluster 和共用儲存體，包括共用儲存體考量，以及如何將受管儲存體轉換為外部儲存體。

主題

- [AWS ParallelCluster 共用儲存考量](#)
- [將 AWS ParallelCluster 受管儲存體轉換為外部儲存體](#)

AWS ParallelCluster 共用儲存考量

在 中 使用共用儲存時，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，請將共用儲存區段新增至您的組態檔案，並建立或更新叢集。
- 若要移除共用儲存，請從組態檔案移除共用儲存區段，並更新叢集。
- 若要將現有的 AWS ParallelCluster 受管共用儲存取代為新的受管儲存，請變更 / [SharedStorage](#) 的值 [Name](#) 並更新叢集。

Warning

根據預設，當您使用新 [Name](#) 參數執行叢集更新時，會刪除現有的 AWS ParallelCluster 受管儲存體和資料。如果您需要變更 [Name](#) 並保留現有的受管共用儲存資料，請務必先 [DeletionPolicy](#) 將設定為 `Retain` 或備份資料，再更新叢集。

- 如果您未備份 AWS ParallelCluster 受管儲存資料，且 [DeletionPolicy](#) 為 `Delete`，則當您的叢集遭到刪除，或從叢集組態中移除受管儲存並更新叢集時，就會刪除您的資料。
- 如果您未備份 AWS ParallelCluster 受管儲存資料且 [DeletionPolicy](#) 為 `Retain`，則您的檔案系統會在刪除叢集之前分離，並可重新連接至另一個叢集做為外部檔案系統。您的資料會保留。
- 如果從叢集組態中移除 AWS ParallelCluster 受管儲存且 [DeletionPolicy](#) 為 `Retain`，則可以保留叢集資料的外部檔案系統重新連接至叢集。
- 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，會將前端節點 /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體上的現有檔案或目錄。這包括預設使用者的叢集 SSH 金鑰，以防它尚未存在於外部檔案系統上。因此，將相同外部檔案系統掛載到其個別 /home 目錄的所有其他叢集，其叢集預設使用者的 SSH 金鑰也會相同。
- 在將相同外部檔案系統掛載到叢集 /home 目錄的多叢集環境中，只有在第一個叢集將外部檔案系統掛載到 /home 時 AWS ParallelCluster，才會產生 SSH 金鑰，該金鑰會授予在前端節點上建立的運算節點存取權。所有其他叢集都使用相同的 SSH 金鑰。因此，擁有這些共用叢集預設使用者的 SSH 金鑰的任何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 受管儲存體轉換為外部儲存體

了解如何將 AWS ParallelCluster 受管儲存轉換為外部儲存。

這些程序是以下列範例組態檔案程式碼片段為基礎。

```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Delete
...
```

將 AWS ParallelCluster 受管儲存體轉換為外部儲存體

1. 在叢集組態檔案中 Retain 將 DeletionPolicy 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. 從叢集組態檔案移除 SharedStorage 區段。

```
...  
...
```

4. 若要將受 SharedStorage 管變更為外部，SharedStorage 並將其從叢集分離，請執行下列命令。

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. 您的共用儲存體現在位於外部，並與叢集分離。
6. 若要將外部檔案系統連接到原始叢集或其他叢集，請遵循下列步驟。
- a. 取得 FSx for Lustre 檔案系統 ID。

- i. 若要使用 AWS CLI 執行下列命令，並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 管理主控台，請登入並導覽至 <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. 更新檔案系統安全群組規則，以提供對檔案系統和叢集子網路的存取。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群組，以允許傳入和傳出 TCP 流量往返前端節點和運算節點 IP CIDR 範圍或字首。為傳入和傳出 TCP 流量指定 TCP 連接埠 988、1021、1022 和 1023。

如需詳細資訊，請參閱《第 [SharedStorage 2 FsxLustreSettings](#) 版使用者指南》中的 [// FileSystemId](#) 和建立、設定和刪除 Amazon EC2 的安全群組。 [Amazon EC2](#) AWS Command Line Interface

- c. 將 SharedStorage 區段新增至叢集組態。

```
...  
- MountDir: /fsx  
  Name: fsx-external  
  StorageType: FsxLustre  
  FsxLustreSettings:  
    FileSystemId: fs-02e5b4b4abd62d51c  
...
```

- d. 若要將外部共用儲存新增至叢集，請執行下列命令。

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

共用儲存的配額

設定叢集SharedStorage以掛載現有的共用檔案儲存體，並根據下表列出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

檔案共用儲存類型	AWS ParallelCluster 受管儲存	外部儲存	配額淨額總計
Amazon EBS	5	5	5
RAID	1	0	1
Amazon EFS	1	20	21
Amazon FSx †	1 FSx for Lustre	20	21

Note

此配額表已新增至 3.2.0 AWS ParallelCluster 版。

† AWS ParallelCluster 僅支援掛載現有的 Amazon FSx for NetApp ONTAP、Amazon FSx for OpenZFS 和檔案快取系統。它不支援建立新的 FSx for ONTAP、FSx for OpenZFS 和檔案快取系統。

Note

如果您使用 AWS Batch 做為排程器，FSx for Lustre 只能在叢集前端節點上使用。檔案快取不支援 AWS Batch 排程器。

AWS ParallelCluster 資源和標記

透過 AWS ParallelCluster，您可以建立標籤來追蹤和管理 AWS ParallelCluster 資源。您可以定義 AWS CloudFormation 要建立的標籤，並將其傳播到叢集組態檔案中 [Tags 區段](#) 中的所有叢集資源。您也可以使用 AWS ParallelCluster 自動產生的標籤來追蹤和管理資源。

當您建立叢集時，叢集及其資源會以本節中定義的 AWS ParallelCluster 和 AWS 系統標籤進行標記。

AWS ParallelCluster 會將標籤套用至叢集執行個體、磁碟區和資源。若要識別叢集堆疊，會將 AWS 系統標籤 AWS CloudFormation 套用至叢集執行個體。為了識別叢集 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 資源。

Warning

所有 AWS ParallelCluster 標籤都是必要的，不得修改，以避免影響系統功能。因此，您無法修改 AWS 系統標籤。

以下是 AWS ParallelCluster 資源 AWS 的系統標籤範例。

```
"aws:cloudformation:stack-name"="clustername"
```

以下是套用至資源的 AWS ParallelCluster 標籤範例。

```
"parallelcluster:cluster-name"="clustername"
```

您可以在的 Amazon EC2 區段中檢視這些標籤 AWS 管理主控台。

檢視標籤

請完成下列步驟，以在的 Amazon EC2 區段中檢視標籤 AWS 管理主控台。

檢視標籤

1. 導覽位於 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 主控台。
2. 若要檢視所有叢集標籤，請在導覽窗格中選擇標籤。
3. 若要依執行個體檢視叢集標籤，請在導覽窗格中選擇執行個體。
4. 選取叢集執行個體。

5. 選擇執行個體詳細資訊中的管理標籤索引標籤，並檢視標籤。
6. 在執行個體詳細資訊中選擇儲存索引標籤。
7. 選取磁碟區 ID。
8. 在磁碟區中，選擇磁碟區。
9. 選擇磁碟區詳細資訊中的標籤索引標籤，並檢視標籤。

AWS ParallelCluster 前端節點執行個體標籤

金鑰	標籤值
parallelcluster:cluster-name	<i>clustername</i>
Name	HeadNode
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>3.15.0</i>

AWS ParallelCluster 前端節點根磁碟區標籤

標籤鍵	標籤值
parallelcluster:cluster-name	<i>clustername</i>
parallelcluster:node-type	HeadNode

標籤鍵	標籤值
parallelcluster:version	3.15.0

AWS ParallelCluster 運算節點執行個體標籤

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>
aws:ec2launchtemplate:version	1
parallelcluster:node-type	Compute
parallelcluster:queue-name	<i>queue-name</i>
parallelcluster:version	3.15.0

AWS ParallelCluster 運算節點根磁碟區標籤

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	3.15.0

PCUI 標籤

標籤鍵	標籤值
parallelcluster-ui	true

監控 AWS ParallelCluster 和日誌

監控是維護 AWS ParallelCluster 及其他 AWS 解決方案的可靠性、可用性和效能的重要部分。AWS 提供下列監控工具，讓您監看 AWS ParallelCluster、回報錯誤，並適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀表板，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。
- Amazon CloudWatch Logs 可讓您監控、存放和存取來自 Amazon EC2 執行個體、CloudTrail 及其他來源的日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon CloudWatch Logs 使用者指南](#)。
- AWS CloudTrail 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 為無伺服器事件匯流排服務，可讓您輕鬆將應用程式與來自各種來源的資料互相連線。EventBridge 可從您自己的應用程式、Software-as-a-Service(SaaS) 應用程式 AWS 和服務提供即時資料串流，並將該資料路由到 Lambda 等目標。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 CloudWatch Logs 的整合](#)
- [Amazon CloudWatch 儀表板](#)
- [叢集指標的 Amazon CloudWatch 警示](#)
- [AWS ParallelCluster 設定的日誌輪換](#)
- [pcluster CLI 日誌](#)
- [Amazon EC2 主控台輸出日誌](#)

- [擷取 PCUI 和 AWS ParallelCluster 執行時間日誌](#)
- [擷取和保留日誌](#)

與 Amazon CloudWatch Logs 的整合

如需 CloudWatch Logs 的詳細資訊，請參閱 [Amazon CloudWatch Logs 使用者指南](#)。若要設定 CloudWatch Logs 整合，請參閱 [Monitoring](#) 一節。若要了解如何使用將自訂日誌附加至 CloudWatch 組態 `append-config`，請參閱《Amazon [CloudWatch 使用者指南](#)》中的多個 [CloudWatch 代理程式組態檔案](#)。Amazon CloudWatch

Amazon CloudWatch Logs 叢集日誌

系統會為具有名稱的每個叢集建立日誌群組 `/aws/parallelcluster/cluster-name-<timestamp>` (例如 `/aws/parallelcluster/testCluster-202202050215`)。

每個節點上的每個日誌 (如果路徑包含 `*`，則為一組日誌*) 都有一個名為的日誌串流 `{hostname}.{instance_id}.{logIdentifier}`。(例如 `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`。) CloudWatch [代理程式會將日誌資料傳送至 CloudWatch](#)，該代理程式會在所有叢集執行個體 `root` 上執行。

建立叢集時，會建立 Amazon CloudWatch 儀表板。此儀表板可讓您檢閱儲存在 CloudWatch Logs 中的日誌。如需詳細資訊，請參閱 [Amazon CloudWatch 儀表板](#)。

此清單包含可用於平台、排程器和節點之日誌串流的 `logIdentifier` 和路徑。

可用於平台、排程器和節點之日誌串流

平台	排程器	節點	日誌串流
amazon	awsbatc	HeadNc	dcv-authenticator : /var/log/parallelcluster/parallelcluster_dcv_authenticator.log
redhat	slurm		dcv-ext-authenticator : /var/log/parallelcluster/parallelcluster_dcv_connect.log
ubuntu			dcv-agent : /var/log/dcv/agent.*.log
			dcv-xsession : /var/log/dcv/dcv-xsession.*.log
			dcv-server : /var/log/dcv/server.log

平台	排程器	節點	日誌串流
			dcv-session-launcher : /var/log/dcv/sessionlauncher.log Xdcv : /var/log/dcv/Xdcv.*.log cfn-init : /var/log/cfn-init.log Chef-client : /var/log/chef-client.log
amazon	awsbatch	Compute	cloud-init : /var/log/cloud-init.log
redhat	slurm	HeadNode	受監管 : /var/log/supervisord.log
ubuntu			
amazon	slurm	Compute	cloud-init-output : /var/log/cloud-init-output.log
redhat		HeadNode	computemgtd : /var/log/parallelcluster/computemgtd
ubuntu			slurmd : /var/log/slurmd.log slurm_prolog_epilog : /var/log/parallelcluster/slurm_prolog_epilog.log

平台	排程器	節點	日誌串流
amazon	slurm	HeadNode	sssd : /var/log/sssds/sssds.log
redhat			sssd_domain_default : /var/log/sssds/sssds_default.log
ubuntu			pam_ssh_key_generator : /var/log/parallelcluster/pam_ssh_key_generator.log
			clusterstatusmgtd : /var/log/parallelcluster/clusterstatusmgtd
			clustermgtd : /var/log/parallelcluster/clustermgtd
			compute_console_output : /var/log/parallelcluster/compute_console_output
			slurm_resume : /var/log/parallelcluster/slurm_resume.log
			slurm_suspend : /var/log/parallelcluster/slurm_suspend.log
			slurmctld : /var/log/slurmctld.log
			slurm_fleet_status_manager : /var/log/parallelcluster/slurm_fleet_status_manager.log
amazon	awsbatch	ComputeFleet	system-messages : /var/log/messages
redhat	slurm	HeadNode	
ubuntu	awsbatch	ComputeFleet	syslog : /var/log/syslog
	slurm	HeadNode	

使用的叢集中的任務會 AWS Batch 儲存 CloudWatch Logs FAILED 中狀態為 RUNNING、SUCCEEDED 或的任務輸出。日誌群組為 `/aws/batch/job`，日誌串流名稱格式為 `jobDefinitionName/default/ecs_task_id`。根據預設，這些日誌設定為不會過期，但您可以修改保留期間。如需詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的[變更 CloudWatch 日誌中的日誌資料保留期間](#)。

Amazon CloudWatch Logs 建置映像日誌

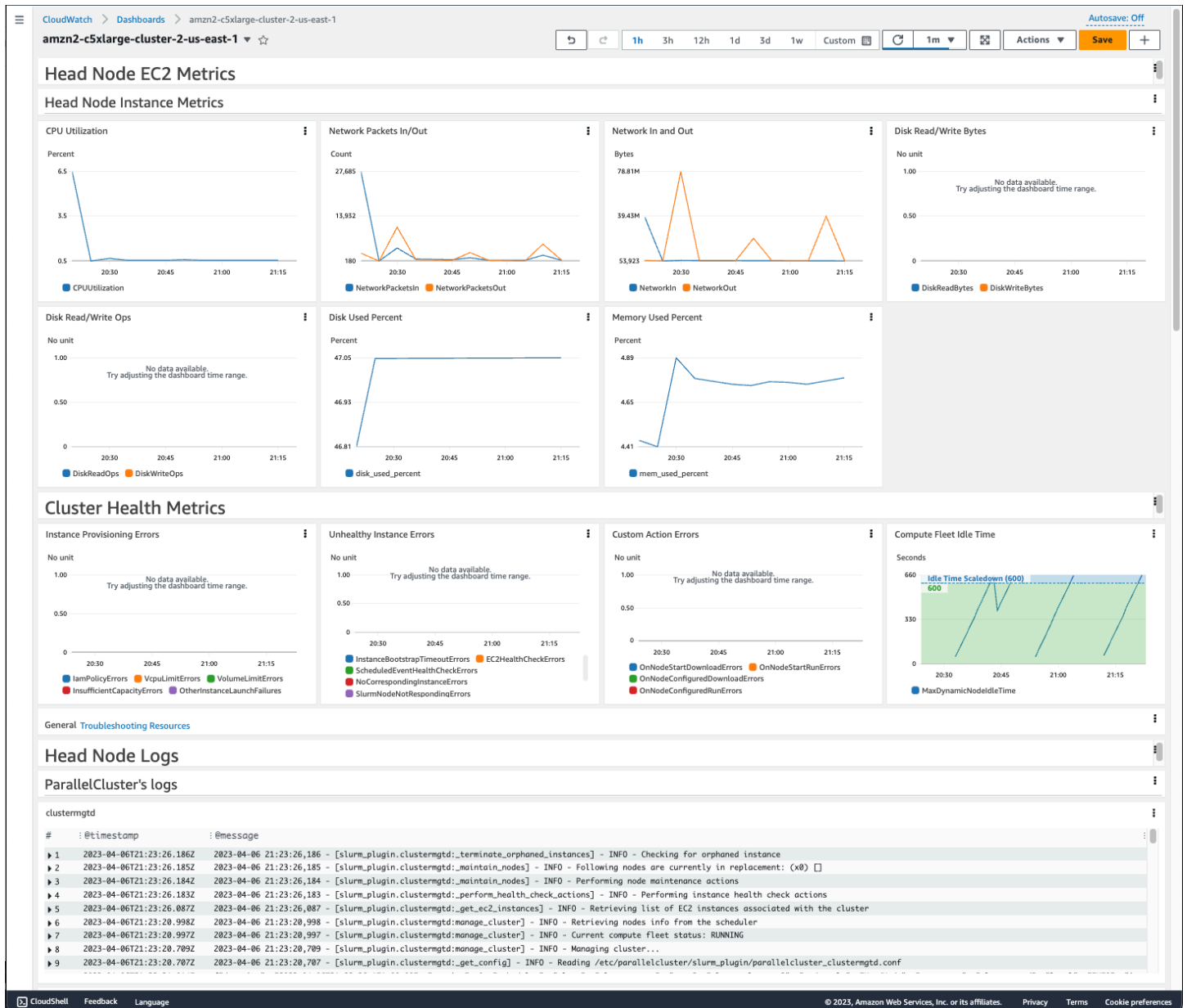
系統會為每個自訂建置映像建立日誌群組，名稱為 `/aws/imagebuilder/ParallelClusterImage-<image-id>`。名稱為 `{pcluster-version}/1` 的唯一日誌串流包含建置映像程序的輸出。

您可以使用 `pcluster` 映像命令存取日誌。如需詳細資訊，請參閱[AWS ParallelCluster AMI 自訂](#)。

Amazon CloudWatch 儀表板

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 CloudWatch Logs 中的日誌。儀表板的名為 `ClusterName-Region`。`ClusterName` 是叢集的名稱 AWS 區域，`# Region` 是叢集所在的區域。您可以在主控台中或透過開啟來存取儀表板 `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`。

下圖顯示叢集的 CloudWatch 儀表板範例。



前端節點執行個體指標

儀表板的第一個區段會顯示前端節點 Amazon EC2 指標的圖形。

如果您的叢集具有共用儲存體，則下一節會顯示共用儲存體指標。

叢集運作狀態指標

如果您的叢集使用 Slurm 進行排程，叢集運作狀態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 [叢集運作狀態指標疑難排解](#)。從 3.6.0 AWS ParallelCluster 版開始，叢集運作狀態指標會新增至儀表板。

前端節點日誌

最後一個區段列出依日誌分組 AWS ParallelCluster 的前端節點日誌、排程器日誌、Amazon DCV 整合日誌和系統日誌。

如需 Amazon CloudWatch 儀表板的詳細資訊，請參閱《[Amazon CloudWatch 使用者指南](#)》中的使用 [Amazon CloudWatch 儀表板](#)。Amazon CloudWatch

如果您不想建立 Amazon CloudWatch 儀表板，您可以將 [Monitoring / Dashboards / CloudWatch](#) / 設定為 [Enabled](#) 將其關閉 `false`。

Note

如果您停用建立 Amazon CloudWatch 儀表板，您也可以停用叢集的 Amazon CloudWatch `disk_used_percent` 和 `memory_used_percent` 警示。如需詳細資訊，請參閱 [叢集指標的 Amazon CloudWatch 警示](#)。

從 3.6 AWS ParallelCluster 版開始新增 `disk_used_percent` 和 `memory_used_percent` 警示。

叢集指標的 Amazon CloudWatch 警示

AWS ParallelCluster 設定 Amazon CloudWatch 警示以監控前端節點的運作狀態和資源使用率。警示名為 `cluster-name-HeadNode-metric`，其中 `cluster-name` 是叢集的名稱，`##` 會識別要監控的指標。

在導覽窗格中選擇警示，以存取 CloudWatch 主控台中的警示。

當觸發任何個別頭部節點警示時，名為 `cluster-name-HeadNode` 的複合警示會進入 ALARM 狀態。

磁碟和記憶體警示

從 3.6.0 AWS ParallelCluster 版開始，會建立下列 CloudWatch 警示：

- `cluster-name-HeadNode-Disk` — 監控根磁碟區 `disk_used_percent` 指標。在 1 分鐘內 1 個資料點的磁碟用量大於 90% 時，進入 ALARM 狀態。
- `cluster-name-HeadNode-Mem` — 監控 `mem_used_percent` 指標。在 1 分鐘內 1 個資料點的記憶體用量大於 90% 時，進入 ALARM 狀態。

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](#)。

運作狀態檢查和 CPU 警示

從 3.8.0 AWS ParallelCluster 版開始，會建立下列 CloudWatch 警示：

- `cluster-name-HeadNode-Health` — 監控 Amazon EC2 StatusCheckFailed 指標。在 1 分鐘內 1 個資料點的值大於 0 時，進入 ALARM 狀態。
- `cluster-name-HeadNode-Cpu` — 監控 Amazon EC2 CPUUtilization 指標。當 CPU 使用率在 1 分鐘內 1 個資料點大於 90% 時，會進入 ALARM 狀態。

叢集管理常駐程式活動訊號警示

從 3.15.0 AWS ParallelCluster 版開始，啟用 Amazon CloudWatch 記錄並使用 Slurm 排程器時，會建立下列警示：

- `cluster-name-HeadNode-ClustermgtdHeartbeat` — 監控 ParallelCluster 命名空間中的 ClustermgtdHeartbeat 指標。在 1 分鐘內連續 10 個資料點收到少於 1 個活動訊號時，警示會進入 ALARM 狀態。遺失的資料會被視為違規。

Note

所有警示會以對稱的方式復原：觸發警示的相同資料點和評估期間也會管理復原。例如，具有 1 個資料點的警示會在相同觀察期間內 1 個良好資料點後復原，同樣地，ClustermgtdHeartbeat 警示需要連續 10 個良好資料點 (10 分鐘) 才能返回 OK。

Note

AWS ParallelCluster 不會設定警示動作。如需有關如何設定警示動作的資訊，例如傳送通知，請參閱 [警示動作](#)。如需 Amazon CloudWatch 警示的詳細資訊，請參閱《[Amazon CloudWatch 使用者指南](#)》中的 [使用 Amazon CloudWatch 警示](#)。Amazon CloudWatch 對於 3.8.0 版和更新 AWS ParallelCluster 版本，請在叢集組態 `false` 中將 [Monitoring / Alarms](#) / 設定為 [Enabled](#) 來停用警示。
對於 3.8.0 之前的 AWS ParallelCluster 版本，請在叢集組態 `false` 中將 [Monitoring / Dashboards / CloudWatch](#) / 設定為 [Enabled](#) 來停用警示。請注意，此設定也會停用 Amazon CloudWatch 儀表板。如需其他詳細資訊 [Amazon CloudWatch 儀表板](#)，請參閱。

AWS ParallelCluster 設定的日誌輪換

AWS ParallelCluster 日誌輪換組態位於 `/etc/logrotate.d/parallelcluster*_log_rotation` 檔案中。當設定的日誌輪換時，目前的日誌內容會保留在單一備份中，而清空的日誌會繼續記錄。

每個設定的日誌只會維護 1 個備份。

AWS ParallelCluster 設定快速增長的日誌，以在大小達到 50 MB 時輪換。快速增長的日誌與擴展和有關 Slurm，包括 `/var/log/parallelcluster/clustermgtd`、`/var/log/parallelcluster/slurm_resume.log` 和 `/var/log/slurmctld.log`。

AWS ParallelCluster 設定在達到 10 MB 大小時旋轉的緩慢成長日誌。

您可以在啟用 CloudFormation [Logs](#) 記錄的情況下，檢視在叢集組態/[CloudWatch/RetentionInDays](#) 設定中定義的天數內保留的較早日誌。檢查 [RetentionInDays](#) 設定，以查看是否需要為您的使用案例增加天數。

AWS ParallelCluster 設定和輪換下列日誌：

標頭節點日誌

```
/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
```

運算節點日誌

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log
```

登入節點日誌

```
/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
```

pcluster CLI 日誌

pcluster CLI 會將命令的日誌寫入 `中的pcluster.log.#檔案/home/user/.parallelcluster/`。

對於每個命令，日誌通常包含具有輸入的命令、用於進行命令的 CLI API 版本副本、回應，以及資訊和錯誤訊息。對於建立和建置命令，日誌也包含組態檔案、組態檔案驗證操作、CloudFormation 範本和堆疊命令。

您可以使用這些日誌來驗證錯誤、輸入、版本和 pcluster CLI 命令。它們也可以做為何時發出命令的記錄。

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，它會嘗試在經過一段時間後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 CloudWatch 通訊，仍可從主控台輸出擷取節點終止原因的實用疑難排解資訊。此主控台輸出會記錄在前端節點的 `/var/log/parallelcluster/compute_console_output` 日誌中。如需 Amazon EC2 主控台輸出的詳細資訊，請參閱《Amazon EC2 Linux [執行個體使用者指南](#)》中的 [執行個體主控台輸出](#)。

根據預設，AWS ParallelCluster 只會從已終止節點的範例子集擷取主控台輸出。這可防止叢集前端節點因大量終止而導致的多個主控台輸出請求負擔過重。根據預設，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。

您可以在前端節點的 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` 檔案中編輯範例大小和等待時間參數值。

此功能已新增至 3.5.0 AWS ParallelCluster 版。

Amazon EC2 主控台輸出參數

您可以在前端節點的 `/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日誌集合，請將 `compute_console_logging_enabled` 設定為 `false`。預設值為 `true`。

您可以隨時更新此參數，而無需停止運算機群。

compute_console_logging_max_sample_size

`compute_console_logging_max_sample_size` 會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，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，而無需停止運算機群。

compute_console_wait_time

`compute_console_wait_time` 會以秒為單位，設定在偵測節點失敗和從該節點收集主控台輸出之間 AWS ParallelCluster 等待的時間。如果您判斷 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，而無需停止運算機群。

擷取 PCUI 和 AWS ParallelCluster 執行時間日誌

了解如何擷取 PCUI 和 AWS ParallelCluster 執行期日誌以進行疑難排解。若要開始，請尋找相關的 PCUI 和 AWS ParallelCluster 堆疊名稱。使用堆疊名稱來尋找安裝日誌群組。若要完成，請匯出日誌。這些日誌專屬於 AWS ParallelCluster 執行時間。如需叢集日誌，請參閱 [擷取和保留日誌](#)。

先決條件

- AWS CLI 已安裝。

- 您有登入資料，可在 AWS 帳戶 PCUI 所在的上執行 AWS CLI 命令。
- 您可以在 PCUI 所在的上存取 AWS 帳戶 Amazon CloudWatch 主控台。

步驟 1：尋找相關堆疊的堆疊名稱

在下列範例中，將紅色反白顯示的文字取代為您的實際值。

使用 AWS 區域 安裝 PCUI 的 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 PCUI 的堆疊名稱。您在安裝 PCUI 時輸入此名稱；例如 `pcluster-ui`。
- 字首為您輸入的 AWS ParallelCluster 堆疊名稱的堆疊，例如 `pcluster-ui-ParallelClusterApi-ABCD1234EFGH`。

步驟 2：尋找日誌群組

列出 PCUI 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
== 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \
  --output text
```

請注意日誌群組的清單，以供下一個步驟使用。

步驟 3：匯出日誌

使用下列步驟來收集和匯出日誌：

1. 登入 AWS 管理主控台，然後導覽至 PCUI 所在的 AWS 帳戶 [Amazon CloudWatch](#) 主控台。
2. 在導覽窗格中選擇日誌、日誌洞見。
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)。

擷取和保留日誌

AWS ParallelCluster 會為 HeadNode 和運算執行個體和儲存體建立 Amazon EC2 指標。您可以在 CloudWatch 主控台自訂儀表板中檢視指標。AWS ParallelCluster 也會在日誌群組中建立叢集 CloudWatch 日誌串流。您可以在 CloudWatch 主控台自訂儀表板或日誌群組中檢視這些日誌。[監控叢集組態區段說明](#)如何修改叢集 CloudWatch 日誌和儀表板。如需詳細資訊，請參閱[與 Amazon CloudWatch Logs 的整合](#)及[Amazon CloudWatch 儀表板](#)。

日誌是疑難排解問題的實用資源。例如，如果您想要刪除失敗的叢集，首先建立叢集日誌的封存可能會有所幫助。遵循 [中的步驟](#) [封存日誌](#) 來建立封存。

主題

- [CloudWatch 中無法使用叢集日誌](#)
- [封存日誌](#)
- [保留的日誌](#)
- [已終止的節點日誌](#)

CloudWatch 中無法使用叢集日誌

如果 CloudWatch 中無法使用叢集日誌，請檢查 ，以確保您在將自訂日誌新增至組態時未覆寫 AWS ParallelCluster CloudWatch 日誌組態。

若要將自訂日誌新增至 CloudWatch 組態，請確定您附加到組態，而不是擷取和覆寫它。如需 `fetch-config` 和 `append-config` 的詳細資訊，請參閱 [《CloudWatch 使用者指南》](#) 中的多個 [CloudWatch 代理程式組態檔案](#)。CloudWatch

若要還原 AWS ParallelCluster CloudWatch 日誌組態，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
```

封存日誌

您可以在 Amazon S3 或本機檔案中封存日誌（取決於 `--output-file` 參數）。

Note

從 AWS ParallelCluster 3.12.0 開始，您可以將日誌匯出至預設 AWS ParallelCluster 儲存貯體。在這種情況下，您不需要設定儲存貯體許可。

Note

將許可新增至 Amazon S3 儲存貯體政策，以授予 CloudWatch 存取。如需詳細資訊，請參閱 [《CloudWatch Logs 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"
}
```

封存包含過去 14 天內來自前端節點和運算節點的 Amazon CloudWatch Logs 串流和 CloudFormation 堆疊事件，除非在組態或 `export-cluster-logs` 命令的參數中明確指定。完成命令所需的時間取決於叢集中的節點數目，以及 CloudWatch Logs 中可用的日誌串流數目。如需可用日誌串流的詳細資訊，請參閱 [與 Amazon CloudWatch Logs 的整合](#)。

保留的日誌

從 3.0.0 版開始，會在刪除叢集時預設 AWS ParallelCluster 保留 CloudWatch Logs。如果您想要刪除叢集並保留其日誌，請確定叢集組態 Delete 中的 [Monitoring / Logs / CloudWatch / DeletionPolicy](#) 未設定為 `None`。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>` 刪除叢集，但保留存放在 Amazon CloudWatch 中的日誌群組。

已終止的節點日誌

如果靜態運算節點意外終止，且 CloudWatch 沒有日誌，請檢查 AWS ParallelCluster 是否已在 `/var/log/parallelcluster/compute_console_output` 日誌中記錄前端節點上該運算節點的主控制台輸出。如需詳細資訊，請參閱 [除錯的金鑰日誌](#)。

如果 `/var/log/parallelcluster/compute_console_output` 日誌無法使用或不包含節點的輸出，請使用從失敗的節點 AWS CLI 擷取主控台輸出。登入叢集前端節點，`instance-id` 並從 `/var/log/parallelcluster/slurm_resume.log` 檔案取得失敗的節點。

使用下列命令搭配 擷取主控台輸出 `instance-id`：

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

如果動態運算節點在啟動後自動終止，且 CloudWatch 沒有日誌，請提交啟用叢集擴展動作的任務。等待執行個體失敗並擷取執行個體主控台日誌。

登入叢集前端節點，`instance-id` 並從 `/var/log/parallelcluster/slurm_resume.log` 檔案取得運算節點。

若要擷取執行個體主控台日誌，請使用下列命令：

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

當運算節點日誌無法使用時，主控台輸出日誌可協助您偵錯運算節點故障的根本原因。

AWS CloudFormation 自訂資源

從 3.6.0 AWS ParallelCluster 版開始，您可以在 AWS CloudFormation 堆疊中使用 an AWS ParallelCluster CloudFormation 自訂資源。自訂資源是 AWS ParallelCluster 託管堆疊。如此一來，您可以使用 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設施。此外，您可以使用 CloudFormation 基礎設施做為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. 新增由 擁有和託管的自訂資源提供者堆疊 AWS ParallelCluster。
2. 參考 CloudFormation 範本中的提供者堆疊做為自訂資源。

自訂資源提供者堆疊會處理和回應 CloudFormation 請求。例如，當您部署 CloudFormation 堆疊時，您也可以設定和建立叢集。若要更新叢集，請更新您的 CloudFormation 堆疊。您可以在刪除堆疊時刪除叢集。如需 CloudFormation 自訂資源的詳細資訊，請參閱AWS CloudFormation 《使用者指南》中的[自訂資源](#)。

Warning

CloudFormation 不會偵測自訂資源偏離。僅使用 CloudFormation 更新叢集組態和刪除叢集。您可以使用 [pcluster](#) CLI 或 [AWS ParallelCluster UI](#) 來監控叢集的狀態，或更新運算機群，但不得使用它們來更新叢集組態或刪除叢集。

Note

建議您將[終止保護](#)新增至堆疊，以避免意外移除。

託管的提供者堆疊 AWS ParallelCluster

自訂資源提供者堆疊的格式如下 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: 3.15.0 }
```

屬性：

參數：

CustomLambdaRole (選用)：

具有執行 AWS Lambda 許可的自訂角色，可建立和管理叢集。根據預設，角色會使用[AWS ParallelCluster 文件中](#)預設定義的相同政策。

AdditionalIamPolicies (選用)：

要新增至 Lambda 使用之角色的其他 IAM 政策 Amazon Resource Name (ARNs) 的逗號分隔清單。只有在 CustomLambdaRole 未指定且可以保留空白時，才會使用此項目。

如果您需要頭部節點、運算節點或存取 Amazon S3 儲存貯體的其他政策，請將它們新增至 CustomLambdaRole 或 AdditionalIamPolicy 屬性。

如果您需要將其他政策連接到前端節點，您還必須授予必要的許可，將這些政策連接到或分離到與前端節點相關聯的 IAM 角色。具體而言，您需要將 "iam : AttachRolePolicy" 和 "iam : DetachRolePolicy" 許可（或其在受管政策中的同等許可）連接至前端節點所使用的 IAM 角色。如需詳細資訊，請參閱[AWS ParallelCluster 用於管理 IAM 資源的使用者範例政策](#)。

如需預設政策的詳細資訊，請參閱 [AWS Identity and Access Management 中的 許可 AWS ParallelCluster](#)。

TemplateURL (必要)：

AWS ParallelCluster 自訂資源檔案 URL。

輸出：**ServiceToken：**

可用作自訂資源ServiceToken屬性的值。自訂資源ServiceToken會指定 CloudFormation 傳送請求的位置。這是您在 CloudFormation 範本中包含之叢集資源的必要輸入。

LogGroupArn：

基礎資源記錄LogGroup的 CloudWatch ARN。

LambdaLayerArn：

用於執行 AWS ParallelCluster 操作的 Lambda 層 ARN。

叢集資源

CloudFormation 叢集資源的格式如下 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：**

AWS ParallelCluster 提供者堆疊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。叢集建立後就無法變更名稱。

ClusterConfiguration：


叢集組態 YAML 檔案，如中所述[叢集組態檔案](#)。不過，您可以使用一般 CloudFormation 建構，例如[內部函數](#)。

DeletionPolicy：

定義是否在刪除根堆疊時刪除叢集。預設值為 Delete。

保留：

如果刪除自訂資源，請保留叢集。

 Note

若要保持保留的叢集正常運作，儲存和聯網等叢集相依資源必須設定要保留的刪除政策。

刪除：

如果刪除自訂資源，請刪除叢集。

Fn::GetAtt 傳回值：

Fn::GetAtt 內部函數會傳回類型之指定屬性的值。如需使用 Fn::GetAtt intrinsic 函數的詳細資訊，請參閱 [Fn::GetAtt](#)。

ClusterProperties：

來自 [pcluster describe-cluster](#) 操作的值。

validationMessages：

字串，包含上次建立或更新操作期間發生的所有驗證訊息。

logGroupName：

用於記錄 Lambda 叢集操作的日誌群組名稱。日誌事件會保留 90 天，而日誌群組會在叢集刪除後保留。

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：簡單、完整的 CloudFormation 範本與 AWS ParallelCluster 自訂資源：

```
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: 3.15.0 }

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
```

若要進一步了解如何使用 CloudFormation AWS ParallelCluster 自訂資源，請參閱 [使用 建立叢集 CloudFormation](#)。

叢集操作

當叢集自訂資源新增至 CloudFormation 堆疊時，CloudFormation 可以執行下列叢集操作：

- CloudFormation 在部署包含 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組態，則根據組態更新政策，CloudFormation 會更新叢集。在更新叢集之前，AWS ParallelCluster 自訂資源提供者不會停止運算機群。建議您使用 [QueueUpdateStrategy](#) 設定進行叢集更新。如此一來，您就可以在使用 AWS ParallelCluster 自訂資源時，避免在更新前後進行明確 `pcluster update-compute-fleet` 呼叫。
- 如果您刪除堆疊，則會刪除叢集。

對包含 AWS ParallelCluster 自訂資源的堆疊進行故障診斷

透過 AWS ParallelCluster 自訂資源，CloudFormation 會從新的個別堆疊部署叢集。您可以執行下列步驟來監控叢集建立：

1. 在 中導覽至 CloudFormation AWS 管理主控台，然後在導覽窗格中選擇 Stacks。
2. 選擇具有您為叢集名稱定義之名稱的堆疊。
3. 如果堆疊狀態為 `ROLLBACK_COMPLETE`，則會在叢集建立期間發生錯誤。
4. 選擇堆疊詳細資訊，然後選擇事件索引標籤。
5. 在您為叢集名稱定義的名稱上搜尋邏輯 ID 上的事件。它具有 `Status reason` 提供問題原因的。

- 您也可以選擇堆疊下拉式選單，然後選擇已刪除以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。
- 若要檢視來自管理叢集之自訂資源提供者的輸出，請選取具有描述「AWS ParallelCluster 叢集自訂資源」的堆疊。選擇資源索引標籤，尋找邏輯 ID 為 `資源PclusterCfnFunctionLogGroup`，然後遵循指定的連結。檢視顯示 Lambda 偵錯輸出的日誌串流。
- 若要對叢集進行疑難排解，請參閱 [AWS ParallelCluster 故障診斷](#)。

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 AWS ParallelCluster 和 Slurm 排程器使用 EFA，請將 [SlurmQueues](#) / [ComputeResources](#) / [Efa](#) / [Enabled](#) 設為 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》中的 [支援的執行個體類型](#)。

我們建議您在置放群組中執行已啟用 EFA 的執行個體。如此一來，執行個體就會在單一可用區域中的低延遲群組中啟動。如需如何使用 設定置放群組的詳細資訊 AWS ParallelCluster，請參閱 [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#)。

Note

不同可用區域不支援 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 分佈會啟用 ptrace (程序追蹤) 保護。ptrace 保護會停用，讓 Libfabric 正常運作。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [停用 ptrace 保護](#)。

預設 EFA 網路組態

從 in AWS ParallelCluster 3.15.0 開始，啟用 EFA 時，AWS ParallelCluster 會自動設定僅限 EFA 的網路介面，將 EFA 流量與 IP 流量分開。這可將 EFA 頻寬最大化，同時將 IP 地址耗用量降至最低。會根據執行個體類型的功能來 AWS ParallelCluster 決定最佳組態。

建議大多數工作負載使用此預設組態，包括緊密耦合的 HPC 和分散式 AI/ML 訓練。

自訂 EFA 網路介面

如果您的工作負載需要不同的網路組態，例如將次要網路卡上的 ENA 頻寬最大化，或設定可用的網路卡子集，您可以使用 [SlurmQueues / ComputeResources / LaunchTemplateOverrides](#) 參數覆寫預設設定。這會將運算節點的整個網路介面組態取代為啟動範本中定義的組態。

如需逐步說明，請參閱[the section called “使用啟動範本覆寫自訂運算節點網路介面”](#)。

Warning

如果您以執行個體類型不支援的方式設定網路介面，執行個體將無法啟動。若要驗證執行個體類型的支援網路組態，請參閱《Amazon EC2 API 參考》中的 [DescribeInstanceTypes](#)。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [Elastic Fabric Adapter](#) 和 [使用彈性布料轉接器擴展 HPC 工作負載](#)，以及 [AWS ParallelCluster](#) AWS 《開放原始碼部落格》中的。

啟用 Intel MPI

Intel MPI 可在 AWS ParallelCluster AMIs 上使用。

Note

若要使用 Intel MPI，您必須確認並接受 [Intel 簡化軟體授權](#) 的條款。

根據預設，Open MPI 會放置在 `path` 上。若要啟用 Intel MPI 而非 Open MPI，您必須先載入 Intel MPI 模組。然後，您需要使用 `安裝最新版本 module load intelmpi`。模組的確切名稱會隨著每次更新而改變。要查看哪些模組可用，請執行 `module avail`。輸出如下。

```
$ module avail
```

```
-----/usr/share/Modules/modulefiles
-----
dot                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
```

要確認 Intel MPI 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 MPI 與 AWS Graviton 型執行個體不相容。

Note

在 2.5.0 AWS ParallelCluster 版之前，Intel MPI 不適用於中國（北京）和中國（寧夏）區域的 AWS ParallelCluster AMIs。

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 AWS 帳戶，即可透過 API 以程式設計方式存取 AWS ParallelCluster 功能。

AWS ParallelCluster API 是以包含 [Amazon API Gateway](#) 端點的獨立 [CloudFormation](#) 範本進行分佈，該端點會公開 AWS ParallelCluster 功能和 [AWS Lambda](#) 函數，負責處理調用的功能。

下圖顯示 AWS ParallelCluster API 基礎設施的高階架構圖。

AWS ParallelCluster API 文件

描述 AWS ParallelCluster API 的 OpenAPI 規格檔案可從下列位置下載：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

從 OpenAPI 規格檔案開始，您可以使用 [Swagger UI](#) 或 [Redoc](#) 等許多可用工具之一來產生 AWS ParallelCluster API 的文件。

如何部署 AWS ParallelCluster API

若要部署 AWS ParallelCluster API，您必須是的管理員 AWS 帳戶。

用於部署 API 的範本可在下列 URL 取得：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

其中 **<REGION>** 是 API 需要部署到 AWS 區域 其中的，而 **<VERSION>** 是 AWS ParallelCluster 版本 (例如 3.15.0)。

AWS Lambda 使用 Lambda layer 界面搭配 [AWS ParallelCluster Python 程式庫 API](#) 來處理 API 調用的功能。

⚠ Warning

中 AWS 帳戶具有 AWS Lambda 或 Amazon API Gateway 服務權限的任何使用者，會自動繼承管理 AWS ParallelCluster API 資源的許可。

使用 部署 AWS ParallelCluster API AWS CLI

在本節中，您將了解如何使用 部署 AWS CLI。

如果您尚未設定要與 CLI 搭配使用的 AWS 登入資料，請加以設定。

```
$ aws configure
```

執行下列命令來部署 API：

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This can be any name
$ VERSION=3.15.0
$ 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 CloudFormation 參數來自訂 API 部署。若要在透過 CLI 部署時設定參數的值，可以使用下列選項：`--parameters ParameterKey=KeyName,ParameterValue=Value`。

下列是選用參數：

- 區域 - 使用 `Region` 參數指定 API 是否能夠控制所有 AWS 區域（預設）或單一中的資源 AWS 區域。將此值設定為正在部署的 AWS 區域 API，以限制存取。
- `ParallelClusterFunctionRole` - 這會覆寫指派給實作功能的 AWS Lambda 函數的 AWS ParallelCluster IAM 角色。參數接受 IAM 角色的 ARN。此角色需要設定為將 AWS Lambda 作為

IAM 主體。此外，由於此角色將取代 API Lambda 函數的預設角色，因此它必須至少具有 API 所需的預設許可，如 中所列 [AWS ParallelCluster 範例 pcluster 使用者政策](#)。

- `ParallelClusterFunctionAdditionalPolicies` - 要連接到 AWS ParallelCluster API 函數角色之其他 IAM 政策的 ARN。只能指定一個政策。
- `CustomDomainName`、`CustomDomainCertificate`、`CustomDomainHostedZoneId` - 使用這些參數來設定 Amazon API Gateway 端點的自訂網域。`CustomDomainName` 是要使用的網域名稱、`CustomDomainCertificate` 此網域名稱的 受管憑證 ARN AWS，以及 `CustomDomainHostedZoneId` 您要建立記錄的 [Amazon Route 53](#) 託管區域 ID。

Warning

您可以設定自訂網域設定，以強制執行 API 的最低版本 Transport Layer Security (TLS)。如需詳細資訊，請參閱 [在 API Gateway 中選擇自訂網域的最低 TLS 版本](#)。

- `EnableIamAdminAccess` - 根據預設，AWS Lambda 處理 AWS ParallelCluster API 操作的函數會設定 IAM 角色，以防止任何特殊權限的 IAM 存取 (`EnableIamAdminAccess=false`)。這使得 API 無法處理需要建立 IAM 角色或政策的操作。因此，只有在資源組態中提供 IAM 角色做為輸入時，才能成功建立叢集或自訂映像。

當 `EnableIamAdminAccess` 設定為 `true` API 時，AWS ParallelCluster 會授予許可，以管理部署叢集或產生自訂 AMIs 所需的 IAM 角色建立。

Warning

將此設為 `true` 時，它會將 IAM 管理員權限授予處理 AWS ParallelCluster 操作的 AWS Lambda 函數。

如需啟用此模式時可解除鎖定功能 [AWS ParallelCluster 用於管理 IAM 資源的使用者範例政策](#) 的其他詳細資訊，請參閱。

- `PermissionsBoundaryPolicy` - 此選用參數接受現有的 IAM 政策 ARN，該政策將設定為 PC API 基礎設施建立的所有 IAM 角色的許可界限，以及管理 IAM 許可的條件，以便只有具有此政策的角色可以由 PC API 建立。

如需此模式所實施限制的其他詳細資訊，請參閱 [PermissionsBoundary 模式](#)。

- `CreateApiUserRole` - 根據預設，AWS ParallelCluster API 的部署包括建立 IAM 角色，該角色設定為授權叫用 API 的唯一角色。Amazon API Gateway 端點已設定 資源型政策，以僅將叫用許可授

予建立的使用者。若要變更此項，請設定 `CreateApiUserRole=false` API 存取權，然後授予選取的 IAM 使用者。如需詳細資訊，請參閱《Amazon API Gateway [API Gateway 開發人員指南](#)》中的[控制叫用 API 的存取](#)。

Warning

當 `CreateApiUserRole=true` API 端點的存取不受 Amazon API Gateway 資源政策限制時，具有無限制 `execute-api:Invoke` 許可的所有 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 資源政策控制對 API 的存取](#)。

Warning

具有叫用所有 AWS ParallelCluster API `ParallelClusterApiUserRole` 操作的許可。若要限制對 API 資源子集的存取，請參閱《[API Gateway 開發人員指南](#)》中的[使用 IAM 政策呼叫 API Gateway API 方法的控制](#)。

- `IAMRoleAndPolicyPrefix` - 此選用參數接受最多包含 10 個字元的字串，做為做為 PC API 基礎設施中建立之 IAM 角色和政策的字首。

更新 API

在本節中，您將了解如何使用兩個可用選項之一來更新 API。

升級到較新的 AWS ParallelCluster 版本

選項 1：若要移除現有的 API，請刪除對應的 CloudFormation 堆疊並部署新的 API，如上所示。

選項 2：若要更新現有的 API，請執行下列命令：

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.15.0
$ 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}
```

叫用 AWS ParallelCluster API

AWS ParallelCluster Amazon API Gateway 端點已使用 [AWS IAM 授權類型](#) 設定，且要求所有請求都必須以有效的 IAM 登入資料簽署 SigV4 ([API 參考：提出 http 請求](#))。

使用預設設定部署時，API 叫用許可只會授予使用 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。

若要擷取 AWS ParallelCluster API 端點，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=='ParallelClusterApiInvokeUrl'].OutputValue" --
output text
```

符合 OpenAPI 規格的任何 HTTP 用戶端都可以叫用 AWS ParallelCluster API，可在這裡找到：

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

請求需要 SigV4 簽署，如 [此處](#) 所述。

目前，我們不提供任何官方 API 用戶端實作。不過，您可以使用 [OpenAPI 產生器](#) 輕鬆從 OpenAPI 模型產生 API 用戶端。產生用戶端後，如果沒有立即提供，則需要新增 SigV4 簽署。

您可以在 [AWS ParallelCluster 儲存庫](#) 中找到 Python API 用戶端的參考實作。若要進一步了解如何使用 Python API 用戶端，請參閱 [使用 AWS ParallelCluster API 教學課程](#)。

若要實作更進階的存取控制機制，例如 Amazon Cognito 或 Lambda 授權方，或使用 AWS WAF 或 API 金鑰進一步保護 API，請遵循 [Amazon API Gateway 文件](#)。

Warning

有權叫用 AWS ParallelCluster API 的 IAM 使用者可以間接控制 AWS ParallelCluster 中管理的所有 AWS 資源 AWS 帳戶。這包括建立使用者因為使用者 IAM 政策的限制而無法直接控制 AWS 的資源。例如，叢集的建立 AWS ParallelCluster 取決於其組態，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 AWS 服務 AWS ParallelCluster 使用者可能無法直接控制的其他資源。

Warning

當您使用組態中 `AdditionalIamPolicies` 指定的 建立叢集時，其他政策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
```

如果您需要其他額外的政策，您可以執行下列其中一項操作：

- 在 `DefaultParallelClusterIamAdminPolicy` 中編輯：


```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/  
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

在 ArnLike/iam:PolicyARN 區段中新增政策。

- 請勿在組態檔案中指定的政策 `AdditionalIamPolicies`，並手動將政策新增至在叢集中建立的 AWS ParallelCluster 執行個體角色。

存取 API 日誌和指標

API 日誌會發佈至 Amazon CloudWatch，保留期為 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 部署相關聯的 Lambda 函數 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 適用於 Terraform

從 AWS ParallelCluster 3.8.0 開始，您可以使用 [Terraform](#) 部署叢集和自訂映像。若要開始使用此功能，請參閱 [Terraform 登錄檔中的 Terraform Provider for AWS ParallelCluster](#)。

Note

您必須在帳戶中部署 [ParallelCluster API](#)，才能使用提供者。

使用以下圖表來判斷提供者與 AWS ParallelCluster 版本之間的相容性：

供應商版本	AWS ParallelCluster 版本
1.0.0	3.8.0-3.10.1
1.1.0	3.11.0+

請參閱如何使用提供者的[範例](#)。

若要獲得更順暢的體驗，請使用 [Terraform Registry AWS 的 ParallelCluster 官方 Terraform 模組](#)。模組可讓您部署：

1. ParallelCluster API
2. 以 YAML 組態檔案和 HCL 定義的 ParallelCluster 叢集
3. ParallelCluster 叢集所需的網路基礎設施

請參閱如何使用 模組的[範例](#)。

透過 Amazon DCV 連接至前端和登入節點

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

Amazon DCV 軟體會自動安裝在前端節點上，並且可以使用 中的 [Dcv](#) 區段來啟用 [HeadNode](#)。

從 3.11 AWS ParallelCluster 版開始，可以使用 / 集區組態中的 Dcv [LoginNodes 區段](#) 區段，為登入節點 [集區](#) 啟用 Amazon DCV。

```
LoginNodes:
  Pools:
    Dcv:
      Enabled: true
```

AWS ParallelCluster 會將 `/home/<DEFAULT_AMI_USER>` 設定為 [DCV 伺服器儲存資料夾](#)。如需 Amazon DCV 組態參數的詳細資訊，請參閱 [HeadNode](#) / [Dcv](#)。

若要連線到前端節點上的 Amazon DCV 工作階段，請使用 [dcv-connect](#) 命令。若要在登入節點上連線，請使用 `dcv-connect` 搭配 `--login-node-ip` 參數，並傳入您要連線之登入節點的公有或私有 IP 地址。

Amazon DCV HTTPS 憑證

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

若要將預設的自我簽署 Amazon DCV 憑證取代為另一個憑證，請先連線到前端節點。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《Amazon DCV 管理員指南》中的 [變更 TLS 憑證](#)。

授權 Amazon DCV

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

AWS ParallelCluster 會自動將必要的許可新增至前端節點 IAM 政策。使用自訂 IAM 執行個體政策時，請使用《[Amazon DCV 管理員指南](#)》中的 [Amazon EC2 上的 Amazon DCV](#) 中所述的許可。

如需疑難排解秘訣，請參閱 [對 Amazon DCV 中的問題進行故障診斷](#)。

使用 `pcluster update-cluster`

In AWS ParallelCluster 3.x 會 [pcluster update-cluster](#) 分析用來建立目前叢集的設定，以及組態檔案中的問題設定。如果發現任何問題，則會回報這些問題，並顯示修正問題所採取的步驟。例如，如果運算 [InstanceType](#) 已變更，則必須先停止運算機群，才能繼續更新。此問題會在發現時回報。如果未發現封鎖問題，則會啟動更新程序並報告變更。

您可以使用 [pcluster update-cluster --dryrun](#) option 在執行之前查看變更。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 故障診斷](#)。

更新政策：定義

更新政策：必須停止叢集中的登入節點，才能變更此設定以進行更新。

當叢集中的登入節點正在使用時，您無法變更這些設定。您必須還原變更，或必須停止叢集登入節點。（您可以將每個集區的計數設定為 0，以停止叢集中的登入節點）。叢集的登入節點停止後，您可以更新叢集（叢集 `update-cluster`）以啟用變更。

Note

從 3.7.0 AWS ParallelCluster 版開始支援此更新政策。

更新政策：可以新增登入節點集區，但移除集區需要停止叢集中的所有登入節點。

若要移除集區，您必須停止叢集中的所有登入節點。（您可以將每個集區的計數設定為 0，以停止叢集中的登入節點）。在叢集的登入節點停止後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟用變更。

Note

從 3.11.0 AWS ParallelCluster 版開始支援此更新政策。

更新政策：必須停止集區中的登入節點，才能變更此設定以進行更新。

當集區中的登入節點正在使用時，您無法變更這些設定。您必須還原變更，或必須停止集區的登入節點。（您可以將集區的計數設定為 0，以停止集區中的登入節點）。在集區的登入節點停止後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟用變更。

Note

從 3.11.0 AWS ParallelCluster 版開始支援此更新政策。

更新政策：此設定可以在更新期間變更。

變更此設定後，可以使用更新叢集 [pcluster update-cluster](#)。

更新政策：如果變更此設定，則不允許更新。

變更此設定後，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的叢集。您可以稍後刪除原始叢集。若要建立新的叢集，請使用 [pcluster create-cluster](#)。若要刪除原始叢集，請使用 [pcluster delete-cluster](#)。

更新政策：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](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

當運算機群存在時，無法變更這些設定。必須還原變更或停止運算機群（使用 [pcluster update-compute-fleet](#)）。運算機群停止後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟用變更。例如，如果您使用Slurm排程器搭配 [SlurmQueues / ComputeResources / - Name / MinCount](#) > 0，則會啟動運算機群。

更新政策：必須停止運算機群和登入節點，才能變更此設定以進行更新。

當運算機群存在或使用登入節點時，無法變更這些設定。必須還原變更或必須停止運算機群和登入節點（可以使用 [停止運算機群pcluster update-compute-fleet](#)）。運算機群和登入節點停止後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟用變更。

更新政策：此設定無法在更新期間減少。

這些設定可以變更，但無法減少。如果必須減少這些設定，您必須還原原始叢集的設定，並使用更新的設定建立新的叢集。您可以稍後刪除原始叢集。若要建立新的叢集，請使用 [pcluster create-cluster](#)。若要刪除原始叢集，請使用 [pcluster delete-cluster](#)。

更新政策：如果變更此設定，則不允許更新。如果您強制更新，則會忽略新值，並使用舊值。

變更此設定後，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的叢集。您可以稍後刪除原始叢集。若要建立新的叢集，請使用 [pcluster create-cluster](#)。若要刪除原始叢集，請使用 [pcluster delete-cluster](#)。

更新政策：必須停止運算機群，或[QueueUpdateStrategy](#)必須設定此設定才能變更更新。

您可以變更這些設定。必須停止（使用 [pcluster update-compute-fleet](#)）或[QueueUpdateStrategy](#)必須設定運算機群。在運算機群停止或設定[QueueUpdateStrategy](#)後，您可以更新叢集 ([pcluster update-cluster](#)) 以啟用變更。

Note


從 3.2.0 AWS ParallelCluster 版開始支援此更新政策。

更新政策：對於此清單值設定，可以在更新期間新增新值，或在移除現有值時必須停止運算機群。

您可以在更新期間新增這些設定的新值。將新值新增至清單後，可以使用 () 更新叢集 [pcluster update-cluster](#)。

若要從清單中移除現有值，必須停止運算機群（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 [Instances/InstanceType](#)，則可以在不停止運算機群的情況下更新叢集。若要從 [Instances/InstanceType](#) 移除現有的執行個體類型，必須先停止運算機群（使用 [pcluster update-compute-fleet](#)）。

 Note

從 3.2.0 AWS ParallelCluster 版開始支援此更新政策。

更新政策：減少佇列的大小需要停止運算機群，或 [QueueUpdateStrategy](#) 必須設定為 TERMINATE，才能針對更新變更此設定。

這些設定可以變更，但如果變更會減少佇列的大小，則必須停止運算機群（使用 [pcluster update-compute-fleet](#)），否則 [QueueUpdateStrategy](#) 必須設定為 TERMINATE。在運算機群停止或 [QueueUpdateStrategy](#) 設定為 TERMINATE 之後，您可以更新叢集 ([叢集 update-cluster](#) 以啟用變更)。

調整叢集容量時設定的 TERMINATE 只會終止節點清單後方的節點，而且會保留相同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]`，在更新期間不會觸碰。只有節點 `st-[4-5]`；`dy-[3-5]` 會在更新期間終止。

支援下列變更，不需要停止運算機群，也不需要將 [QueueUpdateStrategy](#) 設定為 TERMINATE：

- 已新增新的 [SlurmQueue](#)
- 已新增新的 [ComputeResource](#)
- [MaxCount](#) 已增加
- [MinCount](#) 增加，[MaxCount](#) 增加至少相同數量

注意：從 3.9.0 AWS ParallelCluster 版開始，支援此更新政策。

更新政策：對於此清單值設定，運算機群必須停止或[QueueUpdateStrategy](#)必須設定為新增值；在移除現有值時，運算機群必須停止。

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[QueueUpdateStrategy](#)必須設定運算機群。在運算機群停止或設定[QueueUpdateStrategy](#)後，您可以更新叢集（[pcluster update-cluster](#)）以啟用變更。

若要從清單中移除現有值，必須停止運算機群（使用 [pcluster update-compute-fleet](#)）。

Note

從 3.3.0 AWS ParallelCluster 版開始支援此更新政策。

更新政策：所有運算節點都必須停止，才能刪除受管置放群組。必須停止或[QueueUpdateStrategy](#)設定運算機群，才能變更此設定以進行更新。

必須停止運算機群（使用 [pcluster update-compute-fleet](#)），才能移除受管置放群組。如果您在停止運算機群之前執行叢集更新以移除受管置放群組，則會傳回無效的組態訊息，而且不會繼續更新。停止運算機群可確保沒有執行個體正在執行。

pcluster update-cluster 範例

這些設定可以變更，但如果變更會減少佇列的大小，則必須停止運算機群（使用 [pcluster update-compute-fleet](#)），否則 [QueueUpdateStrategy](#) 必須設定為 TERMINATE。在運算機群停止或 [QueueUpdateStrategy](#) 設定為 TERMINATE 之後，您可以更新叢集（[叢集 update-cluster](#)）以啟用變更。

- 此範例示範一些允許變更的更新，並直接開始更新。

```
$ 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": "3.15.0",
    "clusterStatus": "UPDATE_IN_PROGRESS"
```

```

},
"changeSet": [
  {
    "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
    "requestedValue": [
      "sg-0cd61884c4ad11234"
    ],
    "currentValue": [
      "sg-0cd61884c4ad16341"
    ]
  }
]
}
}

```

- 此範例示範一些允許變更的 dryrun 更新。Dryrun 有助於在不啟動更新的情況下報告變更集。

```

$ 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"
      ]
    }
  ]
}
}

```

- 此範例示範 更新，其中包含封鎖更新的一些變更。

```

$ 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 自訂

在某些情況下 AWS ParallelCluster，需要為 建置自訂 AMI。本節涵蓋建置自訂 AWS ParallelCluster AMI 時的考量事項。

您可以使用下列其中一種方法來建置自訂 AWS ParallelCluster AMI：

1. 建立 [建置映像組態檔案](#)，然後使用 pcluster CLI 搭配 EC2 Image Builder 建置映像。此程序為自動化、可重複且支援監控。如需詳細資訊，請參閱 [pcluster](#) 映像命令。
2. 從 AWS ParallelCluster AMI 建立執行個體，然後登入執行個體並進行手動修改。最後，使用 Amazon EC2 從修改的執行個體建立新的 AMI。此程序所需的時間較少。不過，它不是自動化或可重複的，也不支援使用 pcluster CLI 映像監控命令。

如需這些方法的詳細資訊，請參閱 [建置自訂 AWS ParallelCluster AMI](#)。

AWS ParallelCluster AMI 自訂考量事項

無論您如何建立自訂映像，我們都建議您執行初步驗證測試，並包含監控所建立映像狀態的 佈建。

若要使用 建置自訂 AMI pcluster，您可以使用 [EC2 Image Builder](#) 用來 [建置自訂映像的](#) 和 [區段來建立建置映像組態檔案](#)。 [Build ImageBuild](#) 區段指定建置映像所需的映像建置器。這包括 [ParentImage](#) (基礎映像) 和 [Components](#)。 [映像建置器元件](#) 定義在建立映像之前自訂執行個體或測試由建立映像啟動的執行個體所需的一系列步驟。如需詳細資訊，請參閱 EC2 [Image Builder 使用者指南](#) 中的 [使用映像建置器建立自訂元件](#)。

從叢集呼叫 [build-image](#) 以建立自訂映像時，Image Builder 會使用建置映像組態搭配 AWS ParallelCluster 技術指南，在您的 AWS ParallelCluster 上引導 [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。程序完成時，Image Builder 接著會產生新的映像或停止訊息。

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進行測試和驗證。由於映像建置器程序最多可能需要 1 小時，我們建議您事先測試元件。這可以為您節省大量時間。

指令碼案例

在建置映像程序以外的執行中執行個體中測試指令碼，並確認指令碼以結束碼 0 結束。

Amazon Resource Name (ARN) 案例

在建置映像程序之外，在執行中的執行個體中測試元件文件。如需需求清單，請參閱《映像建置器使用者指南》中的[元件管理員](#)。

成功驗證後，將元件新增至建置映像組態

驗證自訂元件是否正常運作後，將其新增至[建置映像組態檔案](#)。

使用 `pcluster` 命令監控映像建置器程序，以協助偵錯

[describe-image](#)

使用此命令來監控建置映像狀態。

[list-image-log-streams](#)

使用此命令來取得日誌串流IDs，您可以使用 `get-image-log-events` 來擷取日誌事件。

[get-image-log-events](#)

使用此命令可取得建置映像程序事件的日誌串流。

例如，您可以使用下列命令來結束建置映像事件。

```
$ 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](#)

使用此命令來擷取映像建置器建立之堆疊的映像堆疊事件。

[export-image-logs](#)

使用此命令儲存映像日誌。

如需 AWS ParallelCluster 日誌和 Amazon CloudWatch 的詳細資訊，請參閱 [Amazon CloudWatch Logs 建置映像日誌](#) 和 [Amazon CloudWatch 儀表板](#)。

其他考量

新 AWS ParallelCluster 版本和自訂 AMIs

如果您建置並使用自訂 AMI，則必須在每個新 AWS ParallelCluster 版本中重複用來建立自訂 AMI 的步驟。

自訂引導操作

檢閱 [自訂引導操作](#) 區段，判斷您欲進行的修改是否可以編寫指令碼，並支援未來的 AWS ParallelCluster 版本。

使用自訂 AMIs

您可以在 [/ CustomAmi](#) 和 [/ Scheduling / SlurmQueues - Image Name/ Image / CustomAmi](#) 區段的叢集組態中指定自訂 AMIs。

若要對自訂 AMI 驗證警告進行疑難排解，請參閱 [對自訂 AMI 問題進行故障診斷](#)。

使用隨需容量預留 (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 執行個體保留容量。如此一來，您就可以獨立於 [Savings Plans](#) 或 [區域預留執行個體](#) 提供的帳單帳戶來建立和管理容量預留。

您可以設定 open 或 targeted ODCR。開啟 ODCR 會涵蓋任何符合 ODCR 屬性的執行個體。這些屬性包括執行個體類型、平台和可用區域。您必須在叢集組態中明確定義目標 ODCR。若要判斷 ODCR 是 open 還是 targeted，請 AWS CLI 執行 Amazon EC2 [describe-capacity-reservation](#) 命令。

您也可以稱之為叢集置放群組 [隨需容量保留 \(CPG ODCR\)](#) 的叢集置放群組中建立 ODCR。

多個 ODCRs 可以在資源群組中分組。這可以在叢集組態檔案中定義。如需資源群組的詳細資訊，請參閱 [《資源群組和標籤使用者指南》中的什麼是資源群組？](#)。

搭配 使用 ODCR AWS ParallelCluster

AWS ParallelCluster 支援開啟的 ODCR。使用開啟的 ODCR 時，您不需要在 中指定任何內容 AWS ParallelCluster。會自動為叢集選取執行個體。您可以指定現有的置放群組，或讓 為您 AWS ParallelCluster 建立新的置放群組。

叢集組態中的 ODCR

從 3.3.0 AWS ParallelCluster 版開始，您可以在叢集組態檔案中定義 ODCRs，而不需要指定 Amazon EC2 run-instances 覆寫。

您可以從使用每個 連結文件中描述的方法建立 [容量保留和資源群組](#) 開始。您必須使用 AWS CLI 方法來建立容量保留群組。如果您使用 AWS 管理主控台，您只能建立標籤型或堆疊型資源群組。使用容量保留啟動執行個體 AWS CLI 時，AWS ParallelCluster 或 不支援標籤型和堆疊型資源群組。

建立容量保留和資源群組之後，請在 [SlurmQueues / CapacityReservationTarget](#) 或 [SlurmQueues // ComputeResources](#) 中指定它們 [CapacityReservationTarget](#)，如下列範例叢集組態所示。以您的有效 # 取代以紅色反白顯示的值。

```
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
```

OBSOLETE/不建議 - 具有 Amazon EC2 執行個體覆寫的目標 ODCR

Warning

- 從 3.3.0 AWS ParallelCluster 版開始，我們不建議使用此方法。本節會保留為使用舊版的實作參考。
- 此方法與 Slurm 的多個執行個體類型配置不相容。

AWS ParallelCluster 3.1.1 中新增了對 targeted ODCRs 支援。在此版本中，推出了一種機制，可覆寫 EC2 RunInstances 參數並傳遞有關保留的資訊，以用於中每個設定的運算資源 AWS ParallelCluster。此機制與 targeted ODCR 相容。不過，當您使用 targeted ODCR 時，您必須指定 run-instances 覆寫組態。必須在 AWS CLI Amazon EC2 [run-instances](#) 命令中明確定義目標 ODCRs。判斷 ODCR 是 open 還是 targeted 執行 AWS CLI Amazon EC2 命令 [describe-capacity-reservation](#)。

多個 ODCRs 可以在資源群組中分組。這可用於 Run-instances 覆寫，以同時鎖定多個 ODCRs。

如果您使用的是 targeted ODCR，您可以指定置放群組。不過，您也需要指定 run-instances 覆寫組態。

假設為您 AWS 建立 targeted ODCR，或者您有一組特定的預留執行個體。然後，您無法指定置放群組。設定的規則 AWS 可能與置放群組設定衝突。因此，如果您的應用程式需要放置群組，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 覆寫組態，而且必須在叢集組態中指定相同的置放群組。

搭配 使用預留執行個體 AWS ParallelCluster

預留執行個體與容量預留 (ODCR) [不同](#)。預留執行個體有 [2 種類型](#)。區域預留執行個體不會保留容量。區域預留執行個體會指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，則不會保留容量，而且可能會收到容量不足的錯誤。如果您有區域預留執行個體，您有容量保留，但沒有可用於指定它們的 run-instances API 參數。

任何 AWS ParallelCluster 版本都支援預留執行個體。您不需要在 `Queue` 中指定任何內容，AWS ParallelCluster 而且會自動選取執行個體。

使用區域預留執行個體時，您可以省略叢集組態中的置放群組規格，以避免潛在的容量不足錯誤。

OBSOLETE/不建議 - 使用 `RunInstances` 自訂 in AWS ParallelCluster 3 進行隨需容量保留 **targeted** (ODCR)

Warning

- 從 3.3.0 AWS ParallelCluster 版開始，我們不建議使用此方法。本節會保留為使用舊版的實作參考。
- 此方法與 Slurm 的多個執行個體類型配置不相容。

您可以覆寫叢集佇列中設定的每個運算資源的 Amazon EC2 RunInstances 參數。若要這樣做，請使用下列程式碼片段內容在叢集的前端節點上建立 `/opt/slurm/etc/pcluster/run_instances_overrides.json` 檔案：

- `${queue_name}` 是您要套用覆寫的佇列名稱。
- `${compute_resource_name}` 是您要套用覆寫的運算資源。
- `${overrides}` 是任意 JSON 物件，其中包含用於特定佇列和執行個體類型組合的 RunInstances 覆寫清單。覆寫語法需要遵循 [Run_instances](#) boto3 呼叫中記錄的相同規格。

```
{
  "${queue_name}": {
    "${compute_resource_name}": {
      ${overrides}
    },
    ...
  },
  ...
}
```

例如，下列 JSON 會將 ODCR 群組設定為 `group_arn` 用於在 `my-queue` 和 `my-compute-resource` 中設定的 `p4d.24xlarge` 執行個體。

```
{
  "my-queue": {
```

```

    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationResourceGroupArn": "group_arn"
        }
      }
    }
  }
}

```

產生此 JSON 檔案後，負責叢集擴展的 AWS ParallelCluster 協助程式會自動使用執行個體啟動的覆寫組態。若要確認指定的參數用於執行個體佈建，請查看下列日誌檔案：

- /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，您會找到包含下列項目的日誌項目：

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

OBSOLETE/不建議 - 使用隨targeted需容量保留 (ODCR) 建立叢集

Warning

- 從 3.3.0 AWS ParallelCluster 版開始，我們不建議使用此方法。本節會保留為使用舊版的實作參考。
- 此方法與 [不相容使用 Slurm 進行多個執行個體類型配置](#)。

1. 建立資源群組，以將容量分組。

```
$ 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

資源群組不支援其他帳戶共用的資源。

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 而不是資源群組。

```
#!/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
```

將容量保留新增至資源群組。每次建立新 ODCR 時，請將其新增至群組保留。*ACCOUNT_ID* 將取代為您的帳戶 ID、將 *PLACEHOLDER_CAPACITY_RESERVATION* 取代為容量保留 ID，並將 *REGION_ID* 取代為 AWS 區域 ID（例如 us-east-1）。

```
$ aws resource-groups group-resources --region REGION_ID --group EC2CRGroup \
  --resource-arns arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-
  reservation/PLACEHOLDER_CAPACITY_RESERVATION
```

在本機電腦上建立政策文件。*ACCOUNT_ID* 將取代為您的帳戶 ID，並將 *REGION_ID* 取代為 AWS 區域 ID（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，在 上建立 IAM 政策。

```

$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-
document file://policy.json

```

3. 在執行個體上於本機建立下列安裝後指令碼，並將其命名為 **postinstall.sh**。

ACCOUNT_ID 將 取代為您的 AWS 帳戶 ID，並將 *REGION_ID* 取代為您的 AWS 區域 ID（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

```

將檔案上傳至 Amazon S3 儲存貯體。將 *amzn-s3-demo-bucket* 取代為您的特定 S3 儲存貯體名稱。

```
$ aws s3 mb s3://amzn-s3-demo-bucket
aws s3 cp postinstall.sh s3://amzn-s3-demo-bucket/postinstall.sh
```

4. 建立本機叢集組態，將預留位置取代為您自己的值。

```
Region: REGION_ID
Image:
  Os: alinux2
HeadNode:
  InstanceType: c5.2xlarge
  Ssh:
    KeyName: YOUR_SSH_KEY
  Iam:
    S3Access:
      - BucketName: amzn-s3-demo-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://amzn-s3-demo-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。

使用下列命令來建立叢集。`cluster-config.yaml` 將取代為您的組態檔案名稱，`cluster-dl` 將取代為您的叢集名稱，將 `REGION_ID` 取代為您的區域 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，安裝後指令碼會在前端節點中執行。指令碼會建立 `run_instances_overrides.json` 檔案並覆寫對的呼叫 `RunInstances`，強制佈建分割區以通過隨需容量保留。

負責叢集擴展的 AWS ParallelCluster 協助程式會自動將此組態用於啟動的新執行個體。若要確認指定參數用於佈建執行個體，您可以查看下列日誌檔案：

- `/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，您會發現日誌項目包含下列項目。

```
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，而無需停止運算機群。套用變更後，所有新的執行個體都會以更新的組態啟動。如果您需要將更新的組態套用至執行中的節點，請強制執行個體終止並等待取代這些節點 AWS ParallelCluster，以回收節點。您可以透過從 Amazon EC2 主控台或終止執行個體 AWS CLI，或將 Slurm 節點設定為 DOWN 或 DRAIN 狀態來執行此操作。

使用以下命令將 Slurm 節點設定為 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 支援機器學習的[隨需容量保留 \(ODCR\)](#) 和容量區塊 (CB)。 [Machine Learning](#) 與 ODCR 不同，CB 可以有未來的開始時間，並且有時間限制。如需使用 ODCR 啟動的詳細資訊，請參閱[使用隨需容量預留 \(ODCR\) 啟動執行個體](#)。

搭配使用 CB AWS ParallelCluster

若要將新的或現有的叢集設定為使用 CB，您必須先在 AWS 帳戶中擁有有效的 CB。您可以使用 AWS 管理主控台 AWS Command Line Interface 或 開發套件，依照官方文件尋找和購買可用的 CB。擁有有效的 CB 後，您可以在 AWS ParallelCluster 組態檔案中設定 CB Amazon Resource Name (ARN) 和相關參數。如需詳細資訊，請參閱[尋找和購買容量區塊 \(CB\)](#)

叢集組態中的 CB

若要針對特定佇列使用 CB，您必須使用 CapacityReservationId 參數。將其設定為現有的 CB ID。您可以從用來建立 CB 的 AWS CLI、AWS 管理主控台或 SDK 取得 CB ARN。

您必須 CapacityType = CAPACITY_BLOCK 為要使用 CB 的佇列設定。將其設定為運算資源 InstanceType 的（與 CB 的 Amazon Elastic Compute Cloud 執行個體類型相同）。

當您 CapacityReservationId 在運算資源層級指定時，InstanceType 是選用的，因為它會自動從保留中擷取。

當您使用時 CapacityType = CAPACITY_BLOCK，MaxCount 必須等於 MinCount 且大於 0，因為屬於 CB 保留的所有執行個體都會以靜態節點進行管理。

在叢集建立時間，前端節點會等待所有靜態節點就緒，再發出叢集建立成功的訊號。不過，當您使用時 CapacityType = CAPACITY_BLOCK，此檢查不會考慮屬於與相關聯之運算資源一部分的節點。即使所有設定的都不是作用中，也會建立叢集。

下列組態檔案程式碼片段顯示在 AWS ParallelCluster 組態檔案中啟用所需的參數。

```
SlurmQueues:
- Name: string
  CapacityType: CAPACITY_BLOCK
  ComputeResources:
  - Name: string
    InstanceType: String (EC2 Instance type of the CB)
    MinCount: integer (<= total capacity of the CB)
    MaxCount: integer (equal to MinCount)
    CapacityReservationTarget:
```

```
CapacityReservationId: String (CB id)
```

AWS ParallelCluster 如何使用容量區塊 (CB)

AWS ParallelCluster 會以特殊方式管理與相關聯的靜態節點。AWS ParallelCluster 即使 CB 尚未處於作用中狀態，仍會建立叢集，並在 CB 處於作用中狀態時自動啟動執行個體。

對應至運算資源、與相關聯且尚未處於作用中狀態的Slurm節點會保留在維護中，直到達到 CB 開始時間為止。Slurm節點會保持保留/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任務，但任務會保留到移除保留pending為止。

AWS ParallelCluster 會自動更新Slurm保留，並將相關的 CB 節點置於維護狀態（對應至 CB 狀態）。當 CB 處於作用中狀態時，會移除Slurm保留，且節點會開始並可用於待定任務或提交新的任務。

達到 CB 結束時間時，節點會移回保留/維護狀態。當 CB 不再作用中且執行個體終止時，最多可以讓使用者將任務重新提交/重新排入佇列至新的佇列/運算資源。

AMI 修補和 Amazon EC2 執行個體替換

為了確保所有動態啟動的叢集運算節點都以一致的方式運作，會 AWS ParallelCluster 停用叢集執行個體自動作業系統更新。此外，會針對每個版本的 AWS ParallelCluster 及其相關聯的 CLI 建置一組特定的 AWS ParallelCluster AMIs。這組特定的 AMIs 保持不變，而且它們僅受其建置的 AWS ParallelCluster 版本支援。發行版本的 AWS ParallelCluster AMIs 不會更新。

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

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

```
$ pcluster version
$ pcluster list-official-images
```

AWS ParallelCluster 前端節點是靜態執行個體，您可以手動更新它。從 3.0.0 AWS ParallelCluster 版開始，完全支援重新啟動和重新啟動前端節點。

如果您的執行個體有暫時性執行個體存放區，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 [LocalStorage](#) 執行個體使用者指南》中的 [HeadNode // EphemeralVolume](#)叢集組態和具有執行個體存放磁碟區的執行個體類型。 <https://>

docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#instance-store-volumes

Amazon EC2

運算節點是暫時性執行個體。根據預設，您只能從前端節點存取它們。從 3.0.0 AWS ParallelCluster 版開始，您可以在使用 [停止運算機群](#) 之後，透過修改 [Scheduling](#) / [SlurmQueues](#) / [Image](#) / [CustomAmi](#) 參數並執行 `pcluster update-cluster` 命令來更新與運算執行個體相關聯的 AMI `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，自動為運算節點建立更新的自訂 AMI：

- 使用 `pcluster build-image` 命令搭配更新的 [建置](#) / [ParentImage](#)。
- 使用 `Build // Enabled : UpdateOsPackages` 執行組建 `true`。

前端節點執行個體更新或取代

在某些情況下，您可能需要重新啟動或重新啟動前端節點。例如，當您手動更新作業系統，或執行個體 [AWS 排程淘汰](#) 強制重新啟動前端節點執行個體時，這是必要的。

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

同樣地，您可以手動停止和啟動沒有執行個體存放區的執行個體。在這種情況下，以及其他沒有暫時性磁碟區的執行個體，請繼續 [停止和啟動叢集的前端節點](#)。

如果您的執行個體有暫時性磁碟機且已停止，執行個體存放區中的資料將會遺失。您可以判斷用於前端節點的執行個體類型是否有執行個體存放區，該存放區位於 [執行個體存放區磁碟區](#) 中的資料表。

從暫時性磁碟機儲存資料

從 3.0.0 AWS ParallelCluster 版開始，每個執行個體類型都完全支援前端節點重新啟動和重新啟動。不過，如果執行個體具有暫時性磁碟機，則會遺失其資料。請依照下列步驟，在頭節點重新啟動或重新開機之前保留您的資料。

若要檢查是否有需要保留的資料，請檢視 / [MountDir](#) 資料夾中的內容 [EphemeralVolume](#) (/scratch 預設為)。

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。

儲存資料之後，請繼續 [停止和啟動叢集的前端節點](#)。

停止和啟動叢集的前端節點

1. 確認叢集中沒有任何執行中的任務。

使用Slurm排程器時：

- 如果未指定 `sbatch--no-requeue` 選項，則會將執行中的任務重新排入佇列。
- 如果指定 `--no-requeue` 選項，則執行中的任務會失敗。

2. 請求叢集運算機群停止：

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOP_REQUESTED",
  ...
}
```

3. 等待運算機群狀態為 STOPPED：

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOPPED",
  ...
}
```

4. 對於作業系統重新啟動或執行個體重新啟動的手動更新，您可以使用 AWS 管理主控台 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. 啟動叢集運算機群：

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
  "status": "START_REQUESTED",
  ...
}
```

作業系統

AWS ParallelCluster 支援 Amazon Linux 2、Amazon Linux 2023、Ubuntu24.04、Ubuntu 22.04、Ubuntu 20.04、Red Hat Enterprise Linux 8 (RHEL8)、Rocky 8、Red Hat Enterprise Linux 9 (RHEL9) 和 Rocky 9。AWS ParallelCluster offers 預先建置特定作業系統的 AMIs，如需提供的 AMIs 詳細資訊，AWS ParallelCluster 請參閱 [Image 區段](#)。

作業系統考量事項

Ubuntu 22.04 和 Ubuntu 24.04

Ubuntu 22.04 和 Ubuntu 24.04 需要更安全的 ssh 金鑰，預設不支援 RSA 金鑰。請產生 ed25519 金鑰，並將其用於建立叢集。

Ubuntu 22.04 無法更新為最新的核心，因為該核心沒有 FSx 用戶端。

RHEL 8

從 3.6.0 AWS ParallelCluster 版開始新增 RedHat Enterprise Linux 8.7 (rhel8)。如果您將叢集設定為使用 rhel8，則任何執行個體類型的隨需成本會高於將叢集設定為使用其他支援的操作系統時。

如需定價的詳細資訊，請參閱 [隨需定價](#) 和 [Amazon Elastic Compute Cloud 上的 Red Hat Enterprise Linux 如何提供和定價？](#)。

Rocky 8

AWS ParallelCluster 3.8.0 支援 Rocky Linux 8，但預先建置的 Rocky Linux 8 AMIs（適用於 x86 和 ARM 架構）無法使用。AWS ParallelCluster 3.8.0 支援使用 [CustomAmi](#) 屬性建立使用自訂 AMIs 的 Rocky Linux 8 叢集。如需建置自訂 AMIs 的詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。

若要從基本 Rocky Linux 8 AMI 建置自訂 AMI，您可以考慮訂閱 AWS [Marketplace](#) 上提供的 [Rocky Linux 8 AMIs](#)。請務必檢閱 AWS Marketplace 上 Rocky Linux 8 AMIs 定價和訂閱成本。或者，您也可以使用官方的 [Rocky Linux 8 AMIs](#) 做為基礎 AMI。

Rocky9

AWS ParallelCluster 3.9.0 支援 Rocky Linux 9，但預先建置的 Rocky Linux 9 AMIs（適用於 x86 和 ARM 架構）無法使用。AWS ParallelCluster 3.9.0 支援使用自訂 AMIs 叢集。[CustomAmi](#) 如需建置自訂 AMIs 的詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。若要從基本 Rocky Linux 9 AMI 建置自訂 AMI，您也可以使用 [官方 Rocky Linux 9 AMIs](#) 做為基本 AMI。如果基本 AMI 沒有最新的核心，自訂 Rocky Linux 9 AMI 建置可能會失敗。在建置 AMI 之前升級核心：

- 從這裡使用 rocky9 AMI ID 啟動執行個體：<https://rockylinux.org/cloud-images/>
- ssh 到執行個體並執行下列命令：`sudo yum -y update`
- 從執行個體建立映像以用作 ParentImage

的參考 AWS ParallelCluster

主題

- [AWS ParallelCluster CLI 命令](#)
- [組態檔案](#)
- [AWS ParallelCluster API 參考](#)
- [AWS ParallelCluster Python 程式庫 API](#)

AWS ParallelCluster CLI 命令

`pcluster` 是主要 AWS ParallelCluster CLI 命令。您可以使用 `pcluster` 在 中啟動和管理 HPC 叢集，AWS 雲端 以及建立和管理自訂 AMI 映像。

`pcluster3-config-converter` 用於將第 2 版格式的 AWS ParallelCluster 叢集組態轉換為第 3 AWS ParallelCluster 版格式。

```
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 AWS ParallelCluster CLI 命令。您可以使用 pcluster 在 中啟動和管理 HPC 叢集 AWS 雲端。

pcluster 會將命令的日誌寫入 中的 `pcluster.log.#檔案/home/user/.parallelcluster/`。如需詳細資訊，請參閱 [pcluster CLI 日誌](#)。

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*

可能的選項：[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-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。

```
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}]
```

具名引數

-h, --help

顯示的說明文字 `pcluster build-image`。

--image-configuration, -c *IMAGE_CONFIGURATION*

將映像組態檔案指定為 YAML 文件。

--image-id, -i *IMAGE_ID*

指定要建置之映像的 ID。

--debug

開啟偵錯記錄

--dryrun *DRYRUN*

當時 `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組態。(預設為 `false`。)

--query *QUERY*

要在輸出上執行的 JMESPath 查詢。

--region, -r *REGION*

指定 AWS 區域要使用的。AWS 區域必須使用映像組態檔案中的 [的區域設定](#)、`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*

當時 `true`，會在失敗時自動啟動映像堆疊轉返。(預設為 `false`。)

--suppress-validators *SUPPRESS_VALIDATORS* [*SUPPRESS_VALIDATORS ...*]

識別要隱藏的一或多個組態驗證器。

格式：`(ALL|type:[A-Za-z0-9]+)`

--validation-failure-level {INFO,WARNING,ERROR}

指定會導致建立失敗的最低驗證層級。(預設為 `ERROR`。)

使用 3.1.2 AWS ParallelCluster 版的範例：

```
$ 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 VPC。如果預設 VPC 已刪除，可能是使用 AWS Control Tower 或 AWS 登陸區域，則必須在映像組態檔案中指定子網路 ID。如需詳細資訊，請參閱 [SubnetId](#)。

pcluster configure

開始第 3 AWS ParallelCluster 版的互動式組態精靈。如需詳細資訊，請參閱 [使用 AWS ParallelCluster 命令列界面設定和建立叢集](#)。

```
pcluster configure [-h]  
    --config CONFIG  
    [--debug]  
    [--region REGION]
```

具名引數

-h, --help

顯示的說明文字 `pcluster configure`。

--config *CONFIG*

輸出產生的組態檔案的路徑。

--debug

開啟偵錯記錄

--region, -r *REGION*

指定 AWS 區域 要使用的。必須使用映像組態檔案中的[區域](#)設定、AWS_DEFAULT_REGION環境變數、~/.aws/config檔案 [default]區段中的region設定或 --region 參數來指定區域。

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}]
```

具名引數

-h, --help

顯示 的說明文字pcluster create-cluster。

--cluster-configuration, -c *CLUSTER_CONFIGURATION*

指定 YAML 叢集組態檔案。

--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。名稱最多可有 60 個字元。如果啟用Slurm會計，名稱最多可有 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*

當時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組態。(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 必須使用叢集組態檔案中[Region](#)的設定、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*

當時true，會在失敗時自動啟動叢集堆疊轉返。(預設為 true。)

--suppress-validators *SUPPRESS_VALIDATORS* [*SUPPRESS_VALIDATORS* ...]

識別要隱藏的一或多個組態驗證器。

格式：(ALL|type : [A-Za-z0-9]+)

--validation-failure-level {INFO,WARNING,ERROR}

指定會導致建立失敗的最低驗證層級。(預設為 ERROR。)

使用 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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

允許使用 Amazon DCV 透過互動式工作階段連線至前端節點。

```
pcluster dcv-connect [-h]
                    --cluster-name CLUSTER_NAME
                    [--debug]
                    [--key-path KEY_PATH]
                    [--login-node-ip LOGIN_NODE_IP]
                    [--region REGION]
                    [--show-url]
```

具名引數

-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。

--login-node-ip

指定叢集中登入節點的公有或私有 IP 地址。使用此引數可讓連線至叢集中已啟用 DCV 的登入節點。

Note

此引數已新增至 3.11.0 AWS ParallelCluster 版。

--region, -r *REGION*

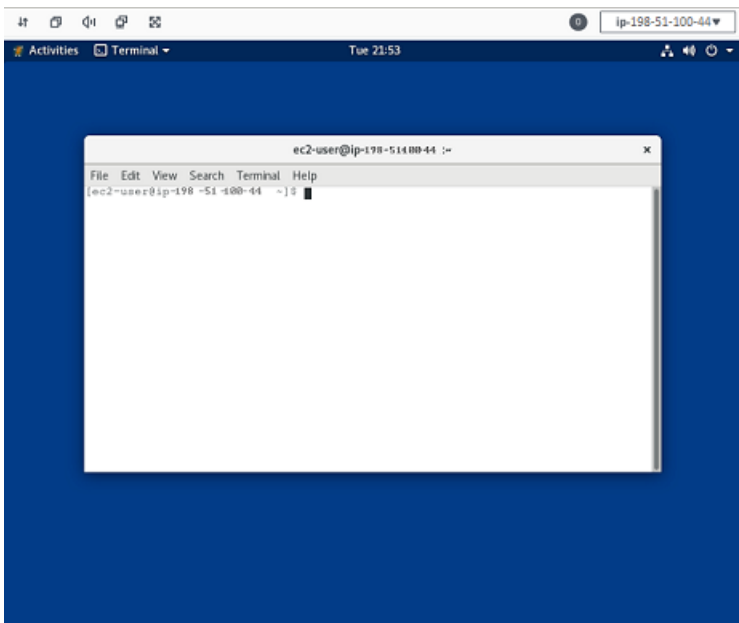
指定 AWS 區域 要使用的。AWS 區域 必須使用 `AWS_DEFAULT_REGION` 環境變數、`~/.aws/config` 檔案 [default] 區段中的 `region` 設定或 `--region` 參數來指定。

--show-url

列印將用於 DCV 連線的 URL 並結束。

使用 3.11 AWS ParallelCluster 版的範例

```
$ pcluster dcv-connect -n cluster-3Dcv --login-node-ip 198.51.100.44 -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]
```

具名引數

-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 [default] 區段中的 `region` 設定或 `--region` 參數來指定區域。

使用 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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

啟動強制終止所有叢集運算節點。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]
```

具名引數

-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*

當時 `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 映像。

```
pcluster delete-image [-h]  
    --image-id IMAGE_ID  
    [--debug]  
    [--force FORCE]  
    [--query QUERY]  
    [--region REGION]
```

具名引數

-h, --help

顯示的說明文字 `pcluster delete-image`。

--image-id, -i *IMAGE_ID*

指定要刪除之映像的 ID。

--debug

啟用偵錯記錄。

--force *FORCE*

當時 `true`，如果有使用 AMI 的執行個體，或共用 AMI，會強制刪除。（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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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]
```

具名引數

-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 3.1.4 AWS ParallelCluster 版的範例：

描述叢集詳細資訊：

```
$ 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",
```



```

    "poolName": "pool1",
    "address": "cluster-v3-eMr9BYRKZVda-e5bb34f40b24f51d.elb.us-
east-1.amazonaws.com",
    "scheme": "internet-facing",
    "healthyNodes": 1,
    "unhealthyNodes": 0
  },
  {
    "status": "active",
    "poolName": "pool2",
    "address": "cluster-v3-PaQ7GgC27sic-aba10c890247b36b.elb.us-
east-1.amazonaws.com",
    "scheme": "internet-facing",
    "healthyNodes": 1,
    "unhealthyNodes": 0
  }
],
"version": "3.1.4",
"clusterConfiguration": {
  "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
},
"tags": [
  {
    "value": "3.11",
    "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"
}

```

使用 `describe-cluster` 擷取叢集組態：

```

$ 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

描述叢集中的執行個體。

```

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]

```

具名引數

-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*

下一組結果的字符。

--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，則會描述 HeadNode、ComputeNode 和 LoginNode 執行個體。

--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，則會描述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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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"
  },
  {
    "launchTime": "2022-07-12T17:37:42.000Z",
    "instanceId": "i-021345abcdef6789",
    "poolName": "pool1",
    "publicIpAddress": "198.51.100.44",
    "instanceType": "t2.micro",
    "state": "pending",
    "nodeType": "loginNode",
    "privateIpAddress": "192.0.2.0.196"
  }
]
}

```

pcluster describe-compute-fleet

描述運算機群的狀態。

```

pcluster describe-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--query QUERY]
    [--region REGION]

```

具名引數

-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 3.1.4 AWS ParallelCluster 版的範例：

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
  "status": "RUNNING",
  "lastStatusUpdateTime": "2022-07-12T17:24:26.000Z"
}
```

pcluster describe-image

取得映像的詳細資訊。

```
pcluster describe-image [-h]
                        --image-id IMAGE_ID
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

具名引數

-h, --help

顯示的說明文字 `pcluster describe-image`。

--image-id, -i *IMAGE_ID*

指定映像的 ID。

--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 3.1.2 AWS ParallelCluster 版的範例：

```
$ 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:amzn-s3-demo-bucket"
      },
      {
        "value": "custom-alinux2-image",
        "key": "parallelcluster:image_name"
      },
      {
        "value": "available",
        "key": "parallelcluster:build_status"
      },
      {
```

```
    "value": "s3://amzn-s3-demo-bucket/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/3.15.0/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

透過傳遞 Amazon S3 儲存貯體，將叢集的日誌匯出至本機tar.gz封存。

```

pcluster export-cluster-logs [-h]
    --cluster-name CLUSTER_NAME
    [--bucket BUCKET_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]
```

Note

`export-cluster-logs` 命令會等待 CloudWatch Logs 完成日誌的匯出，因此預期會經歷一段時間而沒有任何輸出。

具名引數

-h, --help

顯示的說明文字 `pcluster export-cluster-logs`。

--bucket *BUCKET_NAME*

指定要匯出叢集日誌資料的 Amazon S3 儲存貯體名稱。它必須與叢集位於相同的區域。

Note

- 您必須將許可新增至 Amazon S3 儲存貯體政策，才能授予 CloudWatch 存取。如需詳細資訊，請參閱《CloudWatch Logs 使用者指南》中的[設定 Amazon S3 儲存貯體的許可](#)。
- 從 3.12.0 AWS ParallelCluster 版開始，`--bucket` 選項為選用。如果未指定選項，則會使用區域預設儲存 AWS ParallelCluster 貯體 (`parallelcluster-hash-v1-DO-NOT-DELETE`)，或者如果 `CustomS3Bucket` 叢集組態中指定了指向的 Amazon S3 儲存貯體，則會使用該儲存貯體。如果您未指定 `--bucket` 選項並使用預設儲存 AWS ParallelCluster 貯體，則無法將日誌匯出至 `parallelcluster/` 資料夾，因為它是保留供內部使用的受保護資料夾。

Important

如果使用 AWS ParallelCluster 預設儲存貯體，則 `pcluster` 會負責設定儲存貯體政策。如果您自訂儲存貯體政策，然後升級至 AWS ParallelCluster 3.12.0 版，則會覆寫儲存貯體政策，而且您需要重新套用變更。

--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。

根據預設，儲存貯體字首為：

```
cluster-name-logs-202209061743.tar.gz
```

202209061743 是 %Y%m%d%H%M 格式的時間範例。

 Note

從 3.12.0 AWS ParallelCluster 版開始，如果您未指定 `--bucket` 選項並使用預設 AWS ParallelCluster 儲存貯體，則無法將日誌匯出至 `parallelcluster/` 資料夾，因為它是保留供內部使用的受保護資料夾。

--debug

啟用偵錯記錄。

--end-time *END_TIME*

指定收集日誌事件的時間範圍結束，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。時間戳記等於或晚於此時間的事件不會包含在內。可能會省略時間元素（例如分鐘和秒）。預設值是目前時間。

--filters *FILTER* [*FILTER* ...]

指定日誌的篩選條件。格式：`Name=a,Values=1 Name=b,Values=2,3`。支援的篩選條件包括：

`private-dns-name`

指定執行個體私有 DNS 名稱的簡短形式（例如 `ip-10-0-0-101`）。

`node-type`

指定節點類型，此篩選條件唯一接受的值為 `HeadNode`。

--keep-s3-objects *KEEP_S3_OBJECTS*

如果為 true，則會保留匯出至 Amazon S3 的物件。(預設為 false。)

--output-file *OUTPUT_FILE*

指定要儲存日誌封存的檔案路徑。如果提供此功能，則日誌會儲存在本機。否則，它們會以輸出中傳回的 URL 上傳至 Amazon S3。預設為上傳至 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)。包含時間戳記等於或晚於此時間的日誌事件。如果未指定，則預設為建立叢集的時間。

使用 3.1.4 AWS ParallelCluster 版的範例：

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
{
  "url": "https://cluster-v3-bucket..."
}
```

無法擷取日誌？

如果您無法使用 export-cluster-logs 命令擷取日誌，請執行下列其中一項操作：

- 從叢集的 CloudWatch 日誌群組手動擷取日誌。
- 如果日誌群組是空的，SSH 會進入叢集節點，並擷取 中列出的日誌[故障診斷節點初始化問題](#)。
- 如果叢集節點因為叢集無法建立而無法存取，請使用 選項重新建立叢集 --rollback-on-failure false，並從節點擷取日誌。

pcluster export-image-logs

透過傳遞 Amazon S3 儲存貯體，將映像建置器堆疊的日誌匯出至本機 tar.gz 封存。

```
pcluster export-image-logs [-h]
```

```
--image-id IMAGE_ID
[--bucket BUCKET]
[--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]
```

Note

`export-image-logs` 命令會等待 CloudWatch Logs 完成日誌的匯出，因此預期會經歷一段時間而沒有任何輸出。

具名引數

-h, --help

顯示的說明文字 `pcluster export-image-logs`。

--bucket *BUCKET_NAME*

指定要匯出映像建置日誌的 Amazon S3 儲存貯體名稱。它必須與影像位於相同的區域。

Note

- 您必須將許可新增至 Amazon S3 儲存貯體政策，才能授予 CloudWatch 存取。如需詳細資訊，請參閱《CloudWatch Logs 使用者指南》中的[設定 Amazon S3 儲存貯體的許可](#)。
- 從 3.12.0 AWS ParallelCluster 版開始，`--bucket` 選項為選用。如果未指定 選項，則會使用區域預設儲存 AWS ParallelCluster 貯體 (`parallelcluster-hash-v1-D0-NOT-DELETE`)，或如果 `CustomS3Bucket` 已在映像組態中指定，則會使用。

Important

如果使用 AWS ParallelCluster 預設儲存貯體，則 `pcluster` 會負責設定儲存貯體政策。如果您在升級至 3.12.0 AWS ParallelCluster 版之前自訂儲存貯體政策，將會覆寫儲存貯體政策，而且您將需要重新套用變更。

--image-id, -i *IMAGE_ID*

將匯出其日誌的影像 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。

根據預設，儲存貯體字首為：

```
ami-id-logs-202209061743.tar.gz
```

202209061743 是目前時間的 %Y%m%d%H%M 格式。

Note

從 3.12.0 AWS ParallelCluster 版開始，如果您未指定 `--bucket` 選項並使用預設儲存 AWS ParallelCluster 貯體，則無法將日誌匯出至 `parallelcluster/` 資料夾，因為它是保留供內部使用的受保護資料夾。

--debug

啟用偵錯記錄。

--end-time *END_TIME*

指定收集日誌事件的時間範圍結束，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。時間戳記等於或晚於此時間的事件不會包含在內。可能會省略時間元素（例如分鐘和秒）。預設值是目前時間。

--keep-s3-objects *KEEP_S3_OBJECTS*

如果為 `true`，則會保留匯出至 Amazon S3 的物件。（預設為 `false`。）

--output-file *OUTPUT_FILE*

指定要儲存日誌封存的檔案路徑。如果提供此功能，則日誌會儲存在本機。否則，它們會以輸出中傳回的 URL 上傳至 Amazon S3。預設為上傳至 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)。包含時間戳記等於或晚於此時間的日誌事件。如果未指定，則預設為建立叢集的時間。

使用 3.1.4 AWS ParallelCluster 版的範例：

```
$ pcluster export-image-logs --bucket image-v3-bucket --image-id ami-1234abcd5678efgh
{
  "url": "https://image-v3-bucket..."
}
```

pcluster get-cluster-log-events

擷取與日誌串流相關聯的事件。

```
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]
```

具名引數**-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*

指定日誌串流的名稱。您可以使用 list-cluster-log-streams 命令來擷取與事件相關聯的日誌串流。

--debug

啟用偵錯記錄。

--end-time *END_TIME*

指定時間範圍的結束，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ例如 2021-01-01T20:00:00Z)。時間戳記等於或晚於此時間的事件不會包含在內。

--limit *LIMIT*

指定傳回的日誌事件數目上限。如果未指定值，則日誌事件的數量上限為 1 MB 的回應大小，最多 10,000 個日誌事件。

--next-token *NEXT_TOKEN*

下一組結果的字符。

--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，則會先傳回最早的日誌事件。如果值為 false，則會先傳回最新的日誌事件。(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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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

擷取與指定叢集之堆疊相關聯的事件。

Note

從 3.6.0 版開始，AWS ParallelCluster 會使用巢狀堆疊來建立與佇列和運算資源相關聯的資源。GetClusterStackEvents API 和 pcluster get-cluster-stack-events 命令只會傳回叢集主要堆疊事件。您可以在 CloudFormation 主控台中檢視叢集堆疊事件，包括與佇列和運算資源相關的事件。

```
pcluster get-cluster-stack-events [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```


具名引數

-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*

下一組結果的字符。

--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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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

擷取與映像建置相關聯的日誌事件。

```
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]
```

具名引數

-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*

指定日誌串流的名稱。您可以使用 list-image-log-streams 命令來擷取與事件相關聯的日誌串流。

--debug

啟用偵錯記錄。

--end-time *END_TIME*

指定時間範圍的結束，以 ISO 8601 格式表示 (YYYY-MM-DDThh:mm:ssZ 例如 2021-01-01T20:00:00Z)。時間戳記等於或晚於此時間的事件不會包含在內。

--limit *LIMIT*

指定傳回的日誌事件數目上限。如果未指定值，則日誌事件的數量上限為 1 MB 的回應大小，最多 10,000 個日誌事件。

--next-token *NEXT_TOKEN*

下一組結果的字符。

--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`，則會先傳回最早的日誌事件。如果值為 `false`，則會先傳回最新的日誌事件。(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 3.1.2 AWS ParallelCluster 版的範例：

```
$ 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

針對指定的映像建置，擷取與堆疊相關聯的事件。

```
pcluster get-image-stack-events [-h]
    --image-id IMAGE_ID
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

具名引數

-h, --help

顯示的說明文字 pcluster get-image-stack-events。

--image-id, -i *IMAGE_ID*

指定映像的 ID。

--debug

啟用偵錯記錄。

--next-token *NEXT_TOKEN*

下一組結果的字符。

--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 3.1.2 AWS ParallelCluster 版的範例：

```
$ 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]

```

具名引數

-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
```

下一組結果的字符。

```
--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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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

擷取與叢集相關聯的日誌串流清單。

```
pcluster list-cluster-log-streams [-h]
    --cluster-name CLUSTER_NAME
    [--filters FILTERS [FILTERS ...]]
    [--next-token NEXT_TOKEN] [--debug]
    [--query QUERY]
    [--region REGION]
```

具名引數

-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條件。格式：Name=a,Values=1 Name=b,Values=2,3。支援的篩選條件包括：

private-dns-name

指定執行個體私有 DNS 名稱的簡短形式（例如 ip-10-0-0-101）。

node-type

指定節點類型，此篩選條件唯一接受的值為 HeadNode。

--next-token *NEXT_TOKEN*

下一組結果的字符。

--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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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

擷取現有自訂映像的清單。

```
pcluster list-images [-h]  
  --image-status {AVAILABLE,PENDING,FAILED}  
  [--debug]  
  [--next-token NEXT_TOKEN]  
  [--query QUERY]  
  [--region REGION]
```

具名引數

-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*

下一組結果的字符。

--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 3.1.2 AWS ParallelCluster 版的範例：

```
$ 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]
```

具名引數

-h, --help

顯示的說明文字 `pcluster list-image-log-streams`。

--image-id, -i *IMAGE_ID*

指定映像的 ID。

--debug

啟用偵錯記錄。

--next-token *NEXT_TOKEN*

下一組結果的字符。

--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 3.1.2 AWS ParallelCluster 版的範例：

```
$ 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 AWS ParallelCluster AMIs。

```
pcluster list-official-images [-h]  
    [--architecture ARCHITECTURE]  
    [--debug]  
    [--os OS]  
    [--query QUERY]
```

```
[--region REGION]
```

具名引數

-h, --help

顯示的說明文字 `pcluster list-official-images`。

--architecture *ARCHITECTURE*

指定用於篩選結果的架構。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*

指定用於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 必須使用映像組態檔案中的 [區域設定](#)、`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。

使用 3.1.2 AWS ParallelCluster 版的範例：

```
$ pcluster list-official-images
{
  "images": [
    {
      "amiId": "ami-015cfefb4e0d6306b2",
      "os": "ubuntu2204",
      "name": "aws-parallelcluster-3.1.2-ubuntu-2204-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": "ubuntu2204",
    "name": "aws-parallelcluster-3.1.2-ubuntu-2204-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-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

使用預先填入的叢集使用者名稱和 IP 地址執行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]
```

具名引數

-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*

當時 `true`，會列印要執行並結束的命令列。(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
```

使用預先填入之叢集的使用者名稱和 IP 地址執行 `ssh` 命令：

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

pcluster update-cluster

更新現有叢集以符合指定組態檔案的設定。

Note

只有當所有叢集節點都已成功套用更新時，叢集更新才會成功。如果更新失敗，請參閱[使用 `clusterStatus UPDATE_FAILED pcluster describe-cluster` 命令查看](#)一節以取得疑難排解指引。

```
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}]
```

具名引數**-h, --help**

顯示的說明文字 `pcluster update-cluster`。

--cluster-configuration, -c *CLUSTER_CONFIGURATION*

指定 YAML 叢集組態檔案。

--cluster-name, -n *CLUSTER_NAME*

指定叢集的名稱。

--debug

啟用偵錯記錄。

--dryrun *DRYRUN*

當時 `true`，會執行驗證而不更新叢集並建立任何資源。它可用於驗證映像組態和更新需求。(預設為 `false`。)

--force-update *FORCE_UPDATE*

當時 `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 必須使用叢集組態檔案中 [Region](#) 的設定、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 ...*]

識別要隱藏的一或多個組態驗證器。

格式：(ALL|type:[A-Za-z0-9]+)

--validation-failure-level *{INFO,WARNING,ERROR}*

指定回報進行更新的驗證失敗層級。

使用 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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": "amzn-s3-demo-bucket1",
        "KeyName": "output",
        "EnableWriteAccess": false
      }
    },
    {
      "parameter": "HeadNode.Iam.S3Access",
```

```
    "currentValue": {
      "BucketName": "amzn-s3-demo-bucket2",
      "KeyName": "logs",
      "EnableWriteAccess": true
    }
  }
]
```

pcluster update-compute-fleet

更新叢集運算機群的狀態。

```
pcluster update-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    --status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

    [--debug]
    [--query QUERY]
    [--region REGION]
```

具名引數

-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機群的狀態。狀態 START_REQUESTED 和 STOP_REQUESTED 對應至 Slurm 排程器，而狀態 ENABLED 和 DISABLED 對應至 AWS Batch 排程器。

--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 3.1.4 AWS ParallelCluster 版的範例：

```
$ 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]
```

具名引數

-h, --help

顯示 的說明文字 `pcluster version`。

--debug

啟用偵錯記錄。

使用 3.1.4 AWS ParallelCluster 版的範例：

```
$ pcluster version
{
  "version": "3.1.4"
}
```

pcluster3-config-converter

讀取第 2 AWS ParallelCluster 版組態檔案，並寫入第 3 AWS ParallelCluster 版組態檔案。

```
pcluster3-config-converter [-h]
                          [-t CLUSTER_TEMPLATE]
                          [-c CONFIG_FILE]
                          [--force-convert]
                          [-o OUTPUT_FILE]
```

具名引數

-h, --help

顯示的說明文字 pcluster3-config-converter。

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

指定要轉換之組態檔案的 [\[cluster\]](#) 區段。如果未指定，指令碼將在 [\[global\]](#) 區段中尋找 [cluster-template](#) 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 組態檔案。

--force-convert

即使不支援且不建議使用一或多個設定，仍會啟用轉換。

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

指定要寫入的 AWS ParallelCluster 版本 3 組態檔案。如果未指定此參數，則組態會寫入 stdout。

Note

pcluster3-config-converter 命令已在 3.0.1 AWS ParallelCluster 版中新增。

組態檔案

AWS ParallelCluster 使用 YAML 1.1 檔案做為組態參數。

主題

- [叢集組態檔案](#)
- [建置映像組態檔案](#)

叢集組態檔案

AWS ParallelCluster 第 3 版使用不同的組態檔案來控制叢集基礎設施的定義和自訂 AMIs 的定義。所有組態檔案都使用 YAML 1.1 檔案。每個組態檔案的詳細資訊連結如下。如需一些範例組態，請參閱 https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs。

這些物件用於第 3 AWS ParallelCluster 版叢集組態。

主題

- [叢集組態檔案屬性](#)
- [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](#)

叢集組態檔案屬性

Region (選用, String)

指定叢集 AWS 區域的。例如 us-east-2。

更新政策：如果變更此設定，則不允許更新。

CustomS3Bucket (選用, String)

指定在 AWS 帳戶中建立的 Amazon S3 儲存貯體名稱，以存放叢集所使用的資源，例如叢集組態檔案，以及匯出 log。會在您建立叢集的每個 AWS 區域中 AWS ParallelCluster 維護一個 Amazon

S3 儲存貯體。根據預設，這些 Amazon S3 儲存貯體會命名為 `parallelcluster-hash-v1-DO-NOT-DELETE`。

更新政策：如果變更此設定，則不允許更新。如果您強制更新，則會忽略新值，並使用舊值。

AdditionalResources (選用, String)

定義要與叢集一起啟動的其他 AWS CloudFormation 範本。此額外範本用於建立叢集外但屬於叢集生命週期一部分的資源。

此值必須是公有範本的 HTTPS URL，並提供所有參數。

沒有預設值。

更新政策：此設定可以在更新期間變更。

Imds 區段

(選用) 指定全域執行個體中繼資料服務 (IMDS) 組態。

```
Imds:  
  ImdsSupport: string
```

Imds 屬性

ImdsSupport (選用, 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 `ImdsSupport` 設定為 `v1.0`，則同時支援 `IMDSv1` 和 `IMDSv2`。

如果 `ImdsSupport` 設定為 `v2.0`，則僅支援 `IMDSv2`。

如需詳細資訊，請參閱《Amazon EC2 [IMDSv2](#)》。Amazon EC2

更新政策：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`。我們建議您在自訂動作呼叫中 `ImdsSupport` 將設定為 `v2.0`，並將 `IMDSv1` 取代為 `IMDSv2`。

3.3.0 [Imds](#) AWS ParallelCluster 版 [ImdsSupport](#) 新增了對 / 的支援。

Image 區段

Note

由分發的官方 AMIs 不支援版本，在閒置 18 個月後 AWS ParallelCluster 將無法使用。這些舊映像包含過時的軟體，如果發生問題，則無法獲得支援。我們強烈建議您移至最新的支援版本。

(必要) 定義叢集的作業系統。

Image:

Os: *string*

CustomAmi: *string*

Image 屬性

Os (必要, String)

指定要用於叢集的作業系統。支援的值為

alinux2、alinux2023、ubuntu2404、ubuntu2204、rhel8、rocky8rhel9、rocky9。

Note

從 3.6.0 AWS ParallelCluster 版開始新增 RedHat Enterprise 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成本都會高於將叢集設定為使用其他支援的操作系統時。如需定價的詳細資訊，請參閱[隨需定價](#)和 [Amazon EC2 上的 Red Hat Enterprise Linux 如何提供和定價？](#)。

從 3.9.0 AWS ParallelCluster 版開始新增 RedHat Enterprise Linux 9 (rhel9)。

所有 AWS 商業區域都支援下列所有作業系統。

分割區 (AWS 區域)	alinux2	ubuntu224	ubuntu244	rhel8	rhel9	alinux2023
商業 (AWS 區域 未特別提及)	True	True	True	True	True	True

分割區 (AWS 區域)	alinux2	ubuntu224	ubuntu244	rhel8	rhel9	alinux2023
AWS GovCloud (美國東部) (us-gov-east-1)	True	True	True	True	True	True
AWS GovCloud (美國西部) (us-gov-west-1)	True	True	True	True	True	True
中國 (北京) (cn-north-1)	True	True	True	True	True	True
中國 (寧夏) (cn-northwest-1)	True	True	True	True	True	True

更新政策：如果變更此設定，則不允許更新。

Note

AWS ParallelCluster 3.8.0 支援 Rocky Linux 8，但預先建置的 Rocky Linux 8 AMIs (適用於 x86 和 ARM 架構) 不可用。AWS ParallelCluster 3.8.0 支援使用自訂 AMIs 使用 Rocky Linux 8 建立叢集。如需詳細資訊，請參閱 [作業系統考量事項](#)。AWS ParallelCluster 3.9.0 支援 Rocky Linux 9，但預先建置的 Rocky Linux 9 AMIs (適用於 x86 和 ARM 架構) 無法使用。AWS ParallelCluster 3.9.0 支援使用自訂 AMIs 使用 Rocky Linux 9 建立叢集。如需詳細資訊，請參閱 [作業系統考量事項](#)。

CustomAmi (選用, String)

指定要用於前端和運算節點的自訂 AMI ID，而非預設 AMI。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。

如果自訂 AMI 需要其他許可才能啟動，則必須將這些許可同時新增至使用者和前端節點政策。

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

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<AWS_KMS_KEY_ID>"
      ]
    }
  ]
}
```

若要建置 RedHat Enterprise Linux 自訂 AMI，您必須設定作業系統來安裝 RHUI (AWS) 儲存庫提供的套件：`rhel-<version>-baseos-rhui-rpms`、`rhel-<version>-appstream-rhui-rpms` 和 `codeready-builder-for-rhel-<version>-rhui-rpms`。此外，自訂 AMI 上的儲存庫必須包含與執行中核心版本相同版本的 `kernel-devel` 套件。核心。

已知限制：

- 只有 RHEL 8.2 和更新版本支援 FSx for Lustre。
- RHEL 8.7 核心版本 4.18.0-425.3.1.el8 不支援 FSx for Lustre。
- 只有 RHEL 8.4 和更新版本支援 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](#)。

若要對自訂 AMI 驗證警告進行疑難排解，請參閱 [對自訂 AMI 問題進行故障診斷](#)。

[更新政策：如果變更此設定，則不允許更新。](#)

HeadNode 區段

(必要) 指定前端節點的組態。

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
  Imsds:
    Secured: boolean
  Image:
    CustomAmi: string

```

HeadNode 屬性

InstanceType (必要, String)

指定前端節點的執行個體類型。

指定用於前端節點的 Amazon EC2 執行個體類型。執行個體類型的架構必須與用於 [InstanceType](#) 或 AWS Batch Slurm [InstanceType](#) 設定的架構相同。

Note

AWS ParallelCluster 不支援下列執行個體類型的 HeadNode 設定。

- hpc6id

如果您定義具有多個網路介面或網路介面卡的 p4d 執行個體類型或其他執行個體類型，您必須將 `ElasticIp` 設定為 `true` 以提供公有存取。AWS 公有 IPs 只能指派給使用單一網路介面啟動的執行個體。在這種情況下，我們建議您使用 [NAT 閘道](#) 來提供叢集運算節點的公有存取權。如需詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的在執行個體 [啟動期間指派公有 IPv4 地址](#)。

更新政策：如果變更此設定，則不允許更新。

`DisableSimultaneousMultithreading` (選用, Boolean)

如果為 `true`，會在前端節點上停用超執行緒。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》中的 [每個執行個體類型的 CPU 核心和每個 CPU 核心的執行緒](#)。

更新政策：如果變更此設定，則不允許更新。

`SharedStorageType` (選用, 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，以及若未在中指定 [SharedStorage 區段](#) 為掛載目錄以掛載共用檔案系統磁碟區/`/home` 時的預設共用資料。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，則前端節點會將其根磁碟區的一部分匯出為運算節點和使用 NFS 登入節點的共用目錄。

如果為 `Efs`，則 ParallelCluster 會建立 EFS 檔案系統，以用於共用的內部資料和 `/home`。

更新政策：如果變更此設定，則不允許更新。

Note

當叢集向外擴展時，EBS 儲存類型可能會遇到效能瓶頸，因為前端節點會使用 NFS 匯出與運算節點共用根磁碟區中的資料。使用 EFS，您可以避免在叢集橫向擴展時匯出 NFS，並避免與其相關聯的效能瓶頸。建議為小型檔案和安裝程序的最大讀取/寫入潛力選擇 EBS。選擇 EFS 進行擴展。

Networking

(必要) 定義前端節點的聯網組態。

[Networking:](#)

```
SubnetId: string  
ElasticIp: string/boolean  
SecurityGroups:  
  - string  
AdditionalSecurityGroups:  
  - string  
Proxy:  
  HttpProxyAddress: string
```

更新政策：如果變更此設定，則不允許更新。

Networking 屬性

SubnetId (必要, String)

指定要在其中佈建前端節點的現有子網路 ID。

更新政策：如果變更此設定，則不允許更新。

ElasticIp (選用, String)

建立或指派彈性 IP 地址給前端節點。支援的值為 true、false 或現有彈性 IP 地址的 ID。預設值為 false。

更新政策：如果變更此設定，則不允許更新。

SecurityGroups (選用, [String])

要用於前端節點的 Amazon VPC 安全群組 ID 清單。如果不包含此屬性，這些會取代 AWS ParallelCluster 建立的安全群組。

確認您的 [SharedStorage](#) 系統已正確設定安全群組。

更新政策：此設定可以在更新期間變更。

AdditionalSecurityGroups (選用, [String])

要用於前端節點的其他 Amazon VPC 安全群組 ID 清單。

更新政策：此設定可以在更新期間變更。

Proxy (選用)

指定前端節點的代理設定。

```
Proxy:  
HttpProxyAddress:
```

string

HttpProxyAddress (選用, 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。

更新政策：如果變更此設定，則不允許更新。

Ssh

(選用) 定義 SSH 存取前端節點的組態。

Ssh:

KeyName: *string*

AllowedIps: *string*

更新政策：此設定可以在更新期間變更。

Ssh 屬性

KeyName (選用, String)

為現有的 Amazon EC2 金鑰對命名，以啟用對前端節點的 SSH 存取。

更新政策：如果變更此設定，則不允許更新。

AllowedIps (選用, String)

指定 SSH 連線至前端節點的 CIDR 格式 IP 範圍或字首清單 ID。預設值為 `0.0.0.0/0`。

更新政策：此設定可以在更新期間變更。

LocalStorage

(選用) 定義前端節點的本機儲存組態。

LocalStorage:

RootVolume:

Size: *integer*

Encrypted: *boolean*

VolumeType: *string*

```
Iops: integer  
Throughput: integer  
DeleteOnTermination: boolean  
EphemeralVolume:  
MountDir: string
```

更新政策：如果變更此設定，則不允許更新。

LocalStorage 屬性

RootVolume (必要)

指定前端節點的根磁碟區儲存體。

```
RootVolume:  
Size: integer  
Encrypted: boolean  
VolumeType: string  
Iops: integer  
Throughput: integer  
DeleteOnTermination: boolean
```

更新政策：如果變更此設定，則不允許更新。

Size (選用, Integer)

以 GB (GiB) 為單位指定頭部節點根磁碟區大小。預設大小來自 AMI。使用不同的大小需要 AMI 支援 growroot。

更新政策：如果變更此設定，則不允許更新。

Encrypted (選用, Boolean)

指定根磁碟區是否已加密。預設值為 true。

更新政策：如果變更此設定，則不允許更新。

VolumeType (選用, String)

指定 [Amazon EBS 磁碟區類型](#)。支援的值為 gp2、gp3、io1、io2、st1、sc1 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](#)。

更新政策：如果變更此設定，則不允許更新。

Iops (選用, Integer)

定義 io1、io2 和 gp3 類型磁碟區的 IOPS 數目。

預設值、支援的值和 `volume_iops`/`volume_size` 比率因 `VolumeType` 和 `Size` 而異。

更新政策：如果變更此設定，則不允許更新。

`VolumeType = io1`

預設值 Iops = 100

支援的值 Iops = 100–64000 †

最大 Iops/Size 比率 = 每 GiB 50 IOPS。5000 IOPS 需要至少 100 GiB Size 的。

`VolumeType = io2`

預設值 Iops = 100

支援的值 Iops = 100–64000 (io2 區塊快速磁碟區為 256000) †

最大 Iops/Size 比率 = 每 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB Size 的。

`VolumeType = gp3`

預設 Iops = 3000

支援的值 Iops = 3000–16000

最大 Iops/Size 比率 = 每 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB Size 的。

† 只有在佈建超過 32,000 IOPS 的 [Nitro 系統上建置的執行個體上](#)，才能保證 IOPS 上限。其他執行個體保證高達 32,000 IOPS。除非您修改 io1 磁碟區，否則較舊的磁碟區可能無法達到完整效能。 <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modify-volume.html>
io2 Block Express 磁碟區在 R5b 執行個體類型上最多支援 256000 個 Iops 值。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [io2 Block Express 磁碟區](#)。

更新政策：此設定可以在更新期間變更。

Throughput (選用, Integer)

定義 gp3 磁碟區類型的輸送量，以 MiB/s 為單位。此設定僅在 `VolumeType` 為 `gp3` 時有效。預設值為 125。支援的值：125–1000 MiB/s

Throughput 與 `Iops` 的比率不可超過 0.25。1000 MiB/s 的最大輸送量要求 `Iops` 設定至少為 4000。

更新政策：如果變更此設定，則不允許更新。

DeleteOnTermination (選用, Boolean)

指定是否應在前端節點終止時刪除根磁碟區。預設值為 true。

更新政策：如果變更此設定，則不允許更新。

EphemeralVolume (選用)

指定任何執行個體存放區磁碟區的詳細資訊。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[執行個體存放磁碟區](#)。

```
EphemeralVolume:  
MountDir: string
```

更新政策：如果變更此設定，則不允許更新。

MountDir (選用, String)

指定執行個體存放區磁碟區的掛載目錄。預設值為 /scratch。

更新政策：如果變更此設定，則不允許更新。

Dcv

(選用) 定義在前端節點上執行的 Amazon DCV 伺服器的組態設定。

如需詳細資訊，請參閱[透過 Amazon DCV 連接至前端和登入節點](#)。

```
Dcv:  
Enabled: boolean  
Port: integer  
AllowedIps: string
```

Important

根據預設，的 Amazon DCV 連接埠設定 AWS ParallelCluster 開放給所有 IPv4 地址。不過，只有在您擁有 Amazon DCV 工作階段的 URL 並在從傳回 URL 的 30 秒內連線至 Amazon DCV 工作階段時，才能連線至 Amazon DCV 連接埠 `pcluster dcv-connect`。使用 AllowedIps 設定進一步限制存取具有 CIDR 格式 IP 範圍的 Amazon DCV 連接埠，並使用 Port 設定來設定非標準連接埠。

更新政策：如果變更此設定，則不允許更新。

Dcv 屬性

Enabled (必要, Boolean)

指定是否在前端節點上啟用 Amazon DCV。預設值為 false。

更新政策：如果變更此設定，則不允許更新。

Note

Amazon DCV 會自動產生自我簽署憑證，用於保護在前端節點上執行的 Amazon DCV 用戶端和 Amazon DCV 伺服器之間的流量。若要設定您自己的憑證，請參閱[Amazon DCV HTTPS 憑證](#)。

Port (選用, Integer)

指定 Amazon DCV 的連接埠。預設值為 8443。

更新政策：如果變更此設定，則不允許更新。

AllowedIps (選用、建議、String)

指定 Amazon DCV 連線的 CIDR 格式 IP 範圍。只有在 AWS ParallelCluster 建立安全群組時，才會使用此設定。預設值是 0.0.0.0/0，該值允許從任何網際網路位址存取。

更新政策：此設定可以在更新期間變更。

CustomActions

(選用) 指定要在前端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*

OnNodeUpdated:**Sequence:**

- **Script:** *string*

Args:

- *string*

Script: *string*

Args:

- *string*

CustomActions 屬性**OnNodeStart (選用)**

在啟動任何節點部署引導動作之前，指定要在前端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[自訂引導操作](#)。

Sequence (選用)

要執行的指令碼清單。會以與組態檔案中列出的相同順序 AWS ParallelCluster 執行指令碼，從第一個開始。

Script (必要, String)

指定要使用的檔案。檔案路徑的開頭可以是 `https://` 或 `s3://`。

Args (選用, [String])

要傳遞至指令碼的引數清單。

Script (必要, String)

指定要用於單一指令碼的檔案。檔案路徑的開頭可以是 `https://` 或 `s3://`。

Args (選用, [String])

要傳遞至單一指令碼的引數清單。

更新政策：如果變更此設定，則不允許更新。

OnNodeConfigured (選用)

指定節點引導動作完成後，要在前端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[自訂引導操作](#)。

Sequence (選用)

指定要執行的指令碼清單。

Script (必要, String)

指定要使用的檔案。檔案路徑的開頭可以是 https:// 或 s3://。

Args (選用, [String])

要傳遞至指令碼的引數清單。

Script (必要, String)

指定要用於單一指令碼的檔案。檔案路徑的開頭可以是 https:// 或 s3://。

Args (選用, [String])

要傳遞至單一指令碼的引數清單。

[更新政策：如果變更此設定，則不允許更新。](#)

OnNodeUpdated (選用)

指定節點更新動作完成後，要在前端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[自訂引導操作](#)。

Sequence (選用)

指定要執行的指令碼清單。

Script (必要, String)

指定要使用的檔案。檔案路徑的開頭可以是 https:// 或 s3://。

Args (選用, [String])

要傳遞至指令碼的引數清單。

Script (必要, String)

指定要用於單一指令碼的檔案。檔案路徑的開頭可以是 https:// 或 s3://。

Args (選用, [String])

要傳遞至單一指令碼的引數清單。

更新政策：此設定可以在更新期間變更。

Note

OnNodeUpdated 從 AWS ParallelCluster 3.4.0 開始新增。

Sequence 從 3.6.0 AWS ParallelCluster 版開始新增。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，而不包含 Sequence。

AWS ParallelCluster 不支援同時包含單一指令碼和Sequence相同的自訂動作。

Iam

(選用) 指定要在前端節點上使用的執行個體角色或執行個體描述檔，以覆寫叢集的預設執行個體角色或執行個體描述檔。

Iam:

InstanceRole: *string*

InstanceProfile: *string*

S3Access:

- BucketName: *string*

EnableWriteAccess: *boolean*

KeyName: *string*

AdditionalIamPolicies:

- Policy: *string*

更新政策：此設定可以在更新期間變更。

Iam 屬性

InstanceProfile (選用, String)

指定執行個體描述檔以覆寫預設前端節點執行個體描述檔。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`。

如果指定此選項，則無法指定 S3Access 和 AdditionalIamPolicies 設定。

我們建議您指定一個或兩個 S3Access 和 AdditionalIamPolicies 設定，因為新增至的功能 AWS ParallelCluster 通常需要新的許可。

更新政策：如果變更此設定，則不允許更新。

InstanceRole (選用, String)

指定執行個體角色以覆寫預設前端節點執行個體角色。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:role/RoleName`。

如果指定此選項，則無法指定 S3Access 和 AdditionalIamPolicies 設定。

我們建議您指定一個或兩個 S3Access 和 AdditionalIamPolicies 設定，因為新增至 AWS ParallelCluster 的功能通常需要新的許可。

更新政策：此設定可以在更新期間變更。

S3Access

S3Access (選用)

指定儲存貯體。這用於產生政策，以授予儲存貯體的指定存取權。

如果指定此選項，則無法指定 InstanceProfile 和 InstanceRole 設定。

我們建議您指定一個或兩個 S3Access 和 AdditionalIamPolicies 設定，因為新增至的功能 AWS ParallelCluster 通常需要新的許可。

S3Access:

- `BucketName`: *string*
- `EnableWriteAccess`: *boolean*
- `KeyName`: *string*

更新政策：此設定可以在更新期間變更。

BucketName (必要, String)

儲存貯體的名稱。

更新政策：此設定可以在更新期間變更。

KeyName (選用, String)

儲存貯體的金鑰。預設值為 *。

更新政策：此設定可以在更新期間變更。

EnableWriteAccess (選用, Boolean)

指出是否已啟用儲存貯體的寫入存取。預設值為 `false`。

[更新政策：此設定可以在更新期間變更。](#)

AdditionalIamPolicies

AdditionalIamPolicies (選用)

指定 Amazon EC2 IAM 政策的 Amazon Resource Name (ARNs) 清單。除了所需的許可之外，此清單還會連接到用於前端節點的根角色 AWS ParallelCluster。

IAM 政策名稱及其 ARN 不同。無法使用名稱。

如果指定此選項，則無法指定 InstanceProfile 和 InstanceRole 設定。

我們建議您使用 `AdditionalIamPolicies` 因為 `AdditionalIamPolicies` 已新增至 AWS ParallelCluster 所需的許可，且 InstanceRole 必須包含所有必要的許可。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*

[更新政策：此設定可以在更新期間變更。](#)

Policy (選用, [String])

IAM 政策清單。

[更新政策：此設定可以在更新期間變更。](#)

Imsds

(選用) 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](#)。

Imsds:

- Secured: *boolean*

更新政策：如果變更此設定，則不允許更新。

Imds 屬性

Secured (選用, Boolean)

如果為 true，會將對前端節點 IMDS（和執行個體描述檔登入資料）的存取限制為超級使用者的子集。

如果為 false，則前端節點中的每個使用者都可以存取前端節點的 IMDS。

允許下列使用者存取前端節點的 IMDS：

- 根使用者
- 叢集管理使用者 (pc-cluster-admin 預設)
- 作業系統特定的預設使用者 (ec2-user 在 Amazon Linux 2 和 RedHat 上，以及在 Ubuntu 18.04 ubuntu 上)。

預設值為 true。

default 使用者負責確保叢集具有與 AWS 資源互動所需的許可。如果您停用 default 使用者 IMDS 存取，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。

當授予使用者存取主機節點 IMDS 的權限時，他們可以使用[主機節點執行個體描述檔中包含的許可](#)。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分驗證之 AD 網域的密碼。

若要限制 IMDS 存取，AWS ParallelCluster 會管理 鏈 iptables。

具有 sudo 存取權的叢集使用者可以執行 命令，選擇性地啟用或停用其他個別使用者的前端節點 IMDS 存取權，包括 default 使用者：

```
$ 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。

如果您不小心停用 default 使用者 IMDS 存取，您可以使用 --allow 選項還原許可。

Note

任何自訂 iptables 或 ip6tables 規則都會干擾用於限制前端節點上 IMDS 存取的機制。

更新政策：如果變更此設定，則不允許更新。

Image

(選用) 定義前端節點的自訂映像。

```
Image:
  CustomAmi: string
```

更新政策：如果變更此設定，則不允許更新。

Image 屬性

CustomAmi (選用, String)

指定要用於前端節點的自訂 AMI ID，而非預設 AMI。如需詳細資訊，請參閱[AWS ParallelCluster AMI 自訂](#)。

如果自訂 AMI 需要其他許可才能啟動，則必須將這些許可同時新增至使用者和前端節點政策。

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

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<AWS_KMS_KEY_ID>"
      ]
    }
  ]
}
```

若要對自訂 AMI 驗證警告進行疑難排解，請參閱 [對自訂 AMI 問題進行故障診斷](#)。

更新政策：如果變更此設定，則不允許更新。

Scheduling 區段

(必要) 定義叢集中使用的任務排程器，以及任務排程器管理的運算執行個體。您可以使用 Slurm 或 AWS Batch 排程器。每個 都支援不同的設定和屬性集。

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  
LaunchTemplateOverrides:  
  LaunchTemplateId: string  
  Version: 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 (必要, String)

指定使用的排程器類型。支援的值為 slurm 和 awsbatch。

更新政策：如果變更此設定，則不允許更新。

Note

awsbatch 僅支援alinux2作業系統和x86_64平台。

ScalingStrategy (選用, String)

可讓您選擇動態Slurm節點擴展的方式。支援的值為 `all-or-nothing` , `greedy-all-or-nothing` `best-effort` 預設值為 `all-or-nothing`。

[更新政策](#) : 此設定可以在更新期間變更。

Note

擴展策略僅適用於 Slurm 要恢復的節點，不適用於最終已執行的節點。

- `all-or-nothing` 此策略嚴格遵循 `all-or-nothing-approach`，旨在避免擴展程序結束時的閒置執行個體。它以 `all-or-nothing` 為基礎運作，這表示它可以完全擴展或完全不擴展。請注意，當任務需要超過 500 個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的成本。此策略在三種可能的擴展策略中具有最低的輸送量。擴展時間取決於每個 Slurm 繼續程式執行提交的任務數量。此外，您無法擴展超過每次執行的預設 `RunInstances` 資源帳戶限制，預設為 1000 個執行個體。如需更多詳細資訊，請參閱 [Amazon EC2 API 限流文件](#)
- `greedy-all-or-nothing` 與 `all-or-nothing` 策略類似，它旨在避免擴展後的閒置執行個體。此策略允許在擴展過程中暫時過度擴展，以達到高於 `all-or-nothing` 方法的輸送量，但也具有與 `RunInstances` 資源帳戶限制相同的 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可能在擴展程序結束時處於閒置狀態。它會嘗試依任務的要求配置任意數量的節點，但有可能不符合整個請求。與其他策略不同，盡最大努力的方法可以累積比標準 `RunInstances` 限制更多的執行個體，其成本是沿著多個擴展程序執行擁有閒置資源。

每個策略旨在滿足不同的擴展需求，允許您選擇符合您特定需求和限制的項目。

AwsBatchQueues

(選用) 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 `awsbatch` 排程器的詳細資訊，請參閱 [聯網設定](#) 和 [搭配使用 AWS Batch \(awsbatch\) 排程器](#) [AWS ParallelCluster](#)。

[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
```

更新政策：此設定可以在更新期間變更。

AwsBatchQueues 屬性

Name (必要, String)

AWS Batch 佇列的名稱。

更新政策：如果變更此設定，則不允許更新。

CapacityType (選用, String)

AWS Batch 佇列使用的運算資源類型。支援的值為 ONDEMAND、SPOT 或 CAPACITY_BLOCK。預設值為 ONDEMAND。

Note

如果您將 CapacityType 設定為 SPOT，您的帳戶必須包含 AWSServiceRoleForEC2Spot 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 EC2 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](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

Networking

(必要) 定義 AWS Batch 佇列的聯網組態。

```
Networking:  
  SubnetIds:  
    - string  
  AssignPublicIp: boolean  
  SecurityGroups:  
    - string  
  AdditionalSecurityGroups:  
    - string
```

Networking 屬性

SubnetIds (必要, [String])

指定要佈建 AWS Batch 佇列的現有子網路 ID。目前僅支援一個子網路。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

AssignPublicIp (選用, String)

建立或指派公有 IP 地址給 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。

更新政策：如果變更此設定，則不允許更新。

SecurityGroups (選用, [String])

AWS Batch 佇列使用的安全群組清單。如果您未指定安全群組，會 AWS ParallelCluster 建立新的安全群組。

更新政策：此設定可以在更新期間變更。

AdditionalSecurityGroups (選用, [String])

AWS Batch 佇列使用的安全群組清單。

更新政策：此設定可以在更新期間變更。

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 (必要, String)

AWS Batch 佇列運算環境的名稱。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

InstanceTypes (必要, [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。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

MinvCpus (選用, Integer)

AWS Batch 運算環境可以使用 VCPUs 數量下限。

更新政策：此設定可以在更新期間變更。

DesiredVcpus (選用, Integer)

AWS Batch 運算環境中所需的 VCPUs 數量。AWS Batch MaxvCpus 會根據任務佇列中的需求，在 MinvCpus 和 之間調整此值。

更新政策：更新期間不會分析此設定。

MaxvCpus (選用, Integer)

AWS Batch 運算環境 VCPUs 數目上限。您無法將此設定為低於 的值 DesiredVcpus。

更新政策：此設定無法在更新期間減少。

SpotBidPercentage (選用, Float)

在執行個體啟動之前，Amazon EC2 Spot 執行個體價格可以達到的執行個體類型的隨需價格百分比上限。預設值為 100(100%)。支援的範圍為 1-100。

[更新政策](#)：此設定可以在更新期間變更。

SlurmQueues

(選用) 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
  LaunchTemplateOverrides:
    LaunchTemplateId: string
    Version: 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
```

更新政策：對於此清單值設定，可以在更新期間新增新值，或在移除現有值時必須停止運算機群。

SlurmQueues 屬性

Name (必要, 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](#)

更新政策：如果變更此設定，則不允許更新。

CapacityReservationTarget

Note

CapacityReservationTarget 已新增 3.3.0 AWS ParallelCluster 版。

CapacityReservationTarget:

CapacityReservationId: *string*

CapacityReservationResourceGroupArn: *string*

指定佇列運算資源的隨需容量保留。

CapacityReservationId (選用, String)

佇列運算資源目標的現有容量保留 ID。ID 可以參考 [ODCR](#) 或 [ML 的容量區塊](#)。

保留必須使用與執行個體相同的平台。例如，如果您的執行個體在上執行 `rhel8`，您的容量保留必須在 Red Hat Enterprise Linux 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 [支援的平台](#)。

Note

如果您在叢集組態 [Instances](#) 中包含，則必須從組態中排除此佇列層級 CapacityReservationId 設定。

更新政策：必須停止運算機群，或 QueueUpdateStrategy 必須設定此設定才能變更更新。

CapacityReservationResourceGroupArn (選用, String)

做為佇列運算資源之容量保留服務連結群組的資源群組 Amazon Resource Name (ARN)。

AWS ParallelCluster 會根據下列條件，從資源群組識別並使用最適當的容量保留：

- 如果在 [SlurmQueues / Networking](#) 或 [SlurmQueues / ComputeResources](#) / 中啟用 PlacementGroup [Networking](#)，則選取以執行個體類型為目標 AWS ParallelCluster 的資源群組，如果運算資源存在，則 PlacementGroup 選取運算資源的資源群組。

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 / [SlurmQueues Networking](#) 或 [SlurmQueues / ComputeResources](#) / 中未啟用 [Networking](#)，則選擇資源 AWS ParallelCluster 群組，如果運算資源存在，則只會以運算資源的執行個體類型為目標。

資源群組在所有佇列的運算資源和可用區域中，必須在可用區域中為每個預留的執行個體類型至少有一個 ODCR。如需詳細資訊，請參閱[使用隨需容量預留 \(ODCR\) 啟動執行個體](#)。

如需多個子網路組態需求的詳細資訊，請參閱 [Networking / SubnetIds](#)。

Note

3.4.0 AWS ParallelCluster 版中新增了多個可用區域。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

CapacityType (選用, String)

Slurm 佇列使用的運算資源類型。支援的值為 ONDEMAND、SPOT 或 CAPACITY_BLOCK。預設值為 ONDEMAND。

Note

如果您將 CapacityType 設定為 SPOT，您的帳戶必須具有 AWSServiceRoleForEC2Spot 服務連結角色。您可以使用下列 AWS CLI 命令來建立此角色。

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

如需詳細資訊，請參閱《Amazon EC2 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](#)。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

AllocationStrategy (選用, String)

指定中定義之所有運算資源的配置策略 [Instances](#)。

有效值：lowest-price | capacity-optimized | price-capacity-optimized | prioritized | capacity-optimized-prioritized

CapacityType	允許的策略
ONDEMAND	最低價格、優先

CapacityType	允許的策略
SPOT	最低價格、容量最佳化、price-capacity-optimized、capacity-optimized-prioritized
CAPACITY_BLOCK	不支援 — 無法設定 AllocationStrategy

預設 : lowest-price

lowest-price

- 如果您使用 CapacityType = ONDEMAND，Amazon EC2 機群會使用價格來判斷訂單，並先啟動最低價格的執行個體。
- 如果您使用 CapacityType = SPOT，Amazon EC2 機群會從價格最低且具有可用容量的 Spot 執行個體集區啟動執行個體。如果集區在滿足您的所需容量之前用完容量，Amazon EC2 機群會透過為您啟動執行個體來滿足您的請求。特別是，Amazon EC2 機群會從價格最低且具有可用容量的 Spot 執行個體集區啟動執行個體。Amazon EC2 機群可能會從數個不同的集區啟動 Spot 執行個體。
- 如果您設定 CapacityType = CAPACITY_BLOCK，則沒有配置策略，因此無法設定 AllocationStrategy 參數。

capacity-optimized

- 如果您設定 CapacityType = ONDEMAND，capacity-optimized則無法使用。
- 如果您設定 CapacityType = SPOT，Amazon EC2 機群會從 Spot 執行個體集區啟動執行個體，並為要啟動的執行個體數量提供最佳容量。

price-capacity-optimized

- 如果您設定 CapacityType = ONDEMAND，capacity-optimized則無法使用。
- 如果您設定 CapacityType = SPOT，Amazon EC2 機群會識別正在啟動的執行個體數量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 機群會從這些集區的最低價格請求 Spot 執行個體。

prioritized

- 如果您設定 CapacityType = ONDEMAND，Amazon EC2 機群會遵守在指定多個子網路時 AWS ParallelCluster 套用至 LaunchTemplate 覆寫的優先順序。priority 會從中的目標子網路位置 AWS ParallelCluster 衍生覆寫，SlurmQueues/Networking/SubnetIds 第一個子網路的優先順序最高。優先順序由從以 AWS ParallelCluster 遞減順序驅動 SlurmQueues/Networking/SubnetIds，第一個 SubnetId 具有最高優先順序，最後一個 SubnetId 具有最低優先順序。

- 如果您設定 CapacityType = SPOT , prioritized則無法使用。

capacity-optimized-prioritized

- 如果您設定 CapacityType = ONDEMAND , capacity-optimized-prioritized則無法使用。
- 如果您設定 CapacityType = SPOT , Amazon EC2 機群會先最佳化容量，然後盡力套用 AWS ParallelCluster 指派給 LaunchTemplate 覆寫的優先順序。優先順序由從以 AWS ParallelCluster 遞減順序驅動SlurmQueues/Networking/SubnetIds，第一個 SubnetId 具有最高優先順序，最後一個 SubnetID 具有最低優先順序。以相同子網路為目標的所有覆寫都會收到相同的優先順序值。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Note

AllocationStrategy 從 AWS ParallelCluster 3.3.0 版開始支援。
3.14.0 的新功能：prioritized (適用於隨需) 和 capacity-optimized-prioritized (適用於 Spot)。

JobExclusiveAllocation (選用, String)

如果設為 true，Slurm分割區OverSubscribe旗標會設為 EXCLUSIVE。當 OverSubscribe=時EXCLUSIVE，分割區中的任務具有所有配置節點的專屬存取權。如需詳細資訊，請參閱 Slurm 文件中的 [EXCLUSIVE](#)。

有效值：true | false

預設：false

更新政策：此設定可以在更新期間變更。

Note

JobExclusiveAllocation 從 AWS ParallelCluster 3.7.0 版開始支援。

CustomSlurmSettings (選用, 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）的自訂Slurm組態參數鍵值對字典。

每個單獨的鍵值對，例如 Param1: Value1，都會以格式分別新增到Slurm分割區組態列的結尾Param1=Value1。


您只能指定中未拒絕列出的Slurm組態參數CustomSlurmSettings。如需拒絕列出Slurm組態參數的詳細資訊，請參閱 [拒絕列出的Slurm組態參數 CustomSlurmSettings](#)。

AWS ParallelCluster 只會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。您有責任驗證自訂Slurm組態參數。無效的自訂Slurm組態參數可能會導致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Slurm組態參數的詳細資訊，請參閱 Slurm 文件中的 [slurm.conf](#)。

[更新政策：此設定可以在更新期間變更。](#)

 Note

CustomSlurmSettings 從 AWS ParallelCluster 3.6.0 版開始支援。

Tags (選用，【字串】)

標籤鍵/值對的清單。[ComputeResource](#)標籤會覆寫在 [Tags 區段](#)或 / SlurmQueues 中指定的重複標籤Tags。

Key (選用，String)

標籤金鑰。

Value (選用，String)

標籤值。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

HealthChecks (選用)

指定佇列中所有運算資源的運算節點運作狀態檢查。

Gpu (選用)

指定佇列中所有運算資源的 GPU 運作狀態檢查。

Note

AWS ParallelCluster 不支援使用 HealthChecks ARM 作業系統的節點Gpu中的 `alinux2 /`。這些平台不支援 [NVIDIA 資料中心 GPU Manager \(DCGM\)](#)。不建議在使用總 GPU 記憶體大小大於 327680 MiB 的執行個體類型時啟用 GPU 運作狀態檢查。

Enabled (選用, Boolean)

是否在運算節點 AWS ParallelCluster 上執行 GPU 運作狀態檢查。預設值為 `false`。

Gpu 運作狀態檢查行為

- 如果 `Gpu / Enabled` 設定為 `true`，會對佇列中的運算資源 AWS ParallelCluster 執行 GPU 運作狀態檢查。
- `Gpu 運作狀態檢查`會對運算資源執行 GPU 運作狀態檢查，以防止在 GPU 降級的節點上提交任務。
- 如果運算節點未通過`Gpu`運作狀態檢查，則運算節點狀態會變更為 `DRAIN`。新任務不會在此節點上啟動。現有任務會執行至完成。在所有執行中的任務完成後，如果運算節點是動態節點，則運算節點會終止，如果是靜態節點，則會將其取代。
- `Gpu 運作狀態檢查`的持續時間取決於選取的執行個體類型、執行個體中的 GPU 數量、GPU 記憶體總數和`Gpu`運作狀態檢查目標數量（相當於任務 GPU 目標的數量）。例如，在 `p4d.24xlarge` 上，一般持續時間為 3 分鐘。
- 如果`Gpu`運作狀態檢查在不支援的執行個體上執行，它會結束，並在運算節點上執行任務。例如，如果執行個體沒有 GPU，或者如果執行個體有 GPU，但不是 NVIDIA GPU，則運作狀態檢查會結束，且任務會在運算節點上執行。僅支援 NVIDIA GPUs。
- `Gpu 運作狀態檢查`使用 `dcgmi`工具對節點執行運作狀態檢查，並採取下列步驟：

在節點中開始`Gpu`運作狀態檢查時：

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，`Gpu`運作狀態檢查會啟動這些服務。
3. 它會偵測持久性模式是否已啟用。
4. 如果未啟用持久性模式，則`Gpu`運作狀態檢查會啟用該模式。

在運作狀態檢查結束時，`Gpu`運作狀態檢查會將這些服務和資源還原至其初始狀態。

- 如果任務指派給特定節點 GPUs 集，運作Gpu狀態檢查只會在該特定集上執行。否則，Gpu運作狀態檢查會在節點中的所有 GPUs 上執行。
- 如果運算節點同時收到 2 個以上的Gpu運作狀態檢查請求，則只會略過第一個運作狀態檢查和其他項目。這也適用於目標節點 GPUs 的運作狀態檢查。您可以檢查日誌檔案，以取得有關此情況的其他資訊。
- `/var/log/parallelcluster/slurm_health_check.log` 檔案中提供了特定運算節點的運作狀態檢查日誌。檔案可在 Amazon CloudWatch 的叢集 CloudWatch 日誌群組中使用，您可以在其中找到：
 - Gpu 運作狀態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 識別符、序列 ID 和 UUID。
 - 運作狀態檢查輸出。

[更新政策：此設定可以在更新期間變更。](#)

Note

HealthChecks 從 3.6.0 AWS ParallelCluster 版開始支援。

Networking

(必要) 定義Slurm佇列的聯網組態。

```
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
  PlacementGroup:
    Enabled: boolean
    Id: string
    Name: string
  Proxy:
    HttpProxyAddress: string
```

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

Networking 屬性

SubnetIds (必要, [String])

您佈建Slurm佇列的現有子網路 IDs。

如果您在 [SlurmQueues](#) / [ComputeResources](#) / 中設定執行個體類型 [InstanceType](#)，您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，為佇列定義的所有子網路都必須位於相同的 VPC 中，且每個子網路都必須位於個別的可用區域 (AZ) 中。

例如，假設您為佇列定義子網路 1 和子網路 2。

subnet-1 和 subnet-2 不能同時位於 AZ-1 中。

subnet-1 可以在 AZ-1 中，subnet-2 也可以在 AZ-2 中。

如果您只設定一個執行個體類型，並想要使用多個子網路，請在 中定義執行個體類型，Instances 而不是 InstanceType。

例如，定義 `ComputeResources / Instances / InstanceType=instance.type` 而非 `ComputeResources / InstanceType=instance.type`。

Note

不同可用區域不支援 Elastic Fabric Adapter (EFA)。

使用多個可用區域可能會導致儲存聯網延遲增加，並增加可用區域間資料傳輸成本。例如，當執行個體存取位於不同 AZ 的檔案儲存時，可能會發生這種情況。如需詳細資訊，請參閱 [相同 內的資料傳輸 AWS 區域](#)。

從使用單一子網路變更為多個子網路的叢集更新：

- 假設叢集的子網路定義是以單一子網路和 AWS ParallelCluster 受管 FSx for Lustre 檔案系統定義。然後，您無法直接使用更新的子網路 ID 定義更新此叢集。若要進行叢集更新，您必須先將受管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 受管儲存體轉換為外部儲存體](#)。

- 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，如果 EFS 掛載目標不存在於要新增之多個子網路的所有 AZs。然後，您無法直接使用更新的子網路 ID 定義更新此叢集。若要讓叢集更新或建立叢集，您必須先為已定義多個子網路的所有 AZs 建立所有掛載目標。

[CapacityReservationResourceGroupArn](#) 中定義的可用區域和叢集容量保留：

- 如果已定義容量保留資源群組涵蓋的一組執行個體類型和可用區域，以及為佇列定義的一組執行個體類型和可用區域之間沒有重疊，則您無法建立叢集。
- 如果已定義容量保留資源群組涵蓋的一組執行個體類型和可用區域，以及為佇列定義的一組執行個體類型和可用區域之間有部分重疊，您可以建立叢集。會針對此案例 AWS ParallelCluster 傳送有關部分重疊的警告訊息。
- 如需詳細資訊，請參閱[使用隨需容量預留 \(ODCR\) 啟動執行個體](#)。

Note

3.4.0 AWS ParallelCluster 版中新增了多個可用區域。

Warning

如果變更此參數，則此警告不會影響 3.3 AWS ParallelCluster .1 版本 3.3.1 之前的所有 3.x.y AWS ParallelCluster 版本。

3. AWS ParallelCluster 3.1 版之前的 For3 版本：

如果您變更此參數並更新叢集，這會建立新的受管 FSx for Lustre 檔案系統，並刪除現有的受管 FSx for Lustre 檔案系統，而不會保留現有的資料。這會導致資料遺失。在繼續之前，如果您想要保留資料，請務必從現有的 FSx for Lustre 檔案系統備份資料。如需詳細資訊，請參閱《FSx for Lustre 使用者指南》中的[使用備份](#)。

如果新增了新的子網路值，[更新政策：此設定可以在更新期間變更](#)。

如果子網路值已移除，[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新](#)。

AssignPublicIp (選用，String)

建立或指派公有 IP 地址給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。具有公有 IPs子網路預設為指派公有 IP 地址。

如果您定義 p4d 或 hpc6id 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 / ElasticIp](#) 設定為 true，以提供公有存取。AWS 公有 IPs 只能指派給使用單一網路介面啟動的執行個體。在此情況下，我們建議您使用 [NAT 閘道](#) 來提供叢集運算節點的公有存取權。在此情況下，將 AssignPublicIp 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》中的在執行個體 [啟動期間指派公有 IPv4](#) 地址。Amazon EC2

更新政策：如果變更此設定，則不允許更新。

SecurityGroups (選用, [String])

用於 Slurm 佇列的安全群組清單。如果未指定安全群組，會為您 AWS ParallelCluster 建立安全群組。

確認您的 [SharedStorage](#) 系統已正確設定安全群組。

Warning

如果變更此參數，則此警告不會影響 3 AWS ParallelCluster .3.0.3.0 版之前的所有 3.x.y AWS ParallelCluster 版本。

3. AWS ParallelCluster 3.0 版之前的 For3 版本：

如果您變更此參數並更新叢集，這會建立新的受管 FSx for Lustre 檔案系統，並刪除現有的受管 FSx for Lustre 檔案系統，而不會保留現有的資料。這會導致資料遺失。如果您想要保留資料，請務必從現有的 FSx for Lustre 檔案系統備份資料。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 的執行個體是允許所有傳入和傳出流量本身的安全群組成員。

更新政策：此設定可以在更新期間變更。

AdditionalSecurityGroups (選用, [String])

用於 Slurm 佇列的其他安全群組清單。

更新政策：此設定可以在更新期間變更。

PlacementGroup (選用)

指定Slurm佇列的置放群組設定。

```
PlacementGroup:  
  Enabled: boolean  
  Id: string  
  Name: string
```

更新政策：所有運算節點都必須停止，才能刪除受管置放群組。必須停止或QueueUpdateStrategy設定運算機群，才能變更此設定以進行更新。

Enabled (選用, Boolean)

指出置放群組是否用於Slurm佇列。預設值為 false。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Id (選用, String)

Slurm 佇列使用的現有叢集置放群組的置放群組 ID。請務必提供置放群組 ID，而非名稱。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Name (選用, String)

Slurm 佇列使用的現有叢集置放群組的置放群組名稱。請務必提供置放群組名稱，而不是 ID。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Note

- 如果 PlacementGroup / Enabled設定為 true，但未Id定義 Name或，則每個運算資源都會指派自己的受管置放群組，除非 / [Networking](#) / [ComputeResources](#) [PlacementGroup](#)已定義覆寫此設定。
- 從 3.3.0 AWS ParallelCluster 版開始，/ [Networking](#) / [PlacementGroup](#) / [SlurmQueues](#) [Name](#)已新增為 [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#) / 的偏好替代方案[Id](#)。

[PlacementGroup](#) / [Id](#)和 [PlacementGroup](#) / [Name](#)相等。您可以使用其中一個。

如果您同時包含 [PlacementGroup](#) / [Id](#)和 [PlacementGroup](#) / [Name](#)，則 AWS ParallelCluster 失敗。您只能選擇其中一個。

您不需要更新叢集即可使用 [PlacementGroup / Name](#)。

- 使用容量區塊保留時，不應設定置放群組限制，因為即使容量保留有剩餘容量，仍可能因保留之外的置放限制而發生容量不足錯誤。

Proxy (選用)

指定Slurm佇列的代理設定。

```
Proxy:  
HttpProxyAddress: string
```

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

HttpProxyAddress (選用, String)

定義Slurm佇列的 HTTP 或 HTTPS 代理伺服器。一般而言，它是 `https://x.x.x.x:8080`。

沒有預設值。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Image

(選用) 指定要用於Slurm佇列的映像。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
```

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Image 屬性

CustomAmi (選用, String)

用於Slurm佇列的 AMI，而非預設 AMIs。您可以使用 `pcluster CLI` 命令來檢視預設 AMIs 的清單。

Note

AMI 必須基於前端節點所使用的相同作業系統。

```
pcluster list-official-images
```

如果自訂 AMI 需要其他許可才能啟動，您必須將這些許可新增至前端節點政策。

例如，如果自訂 AMI 具有與其相關聯的加密快照，則在前端節點政策中需要以下其他政策。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<AWS_KMS_KEY_ID>"
      ]
    }
  ]
}
```

若要對自訂 AMI 驗證警告進行疑難排解，請參閱 [對自訂 AMI 問題進行故障診斷](#)。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

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](#)。
- 只有在叢集部署在屬於叢集建立時存在之 CIDR 區塊的子網路中時，才能將新的運算資源新增至叢集。

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
  LaunchTemplateOverrides:
    LaunchTemplateId: string
    Version: string

```

更新政策：對於此清單值設定，可以在更新期間新增新值，或在移除現有值時必須停止運算機群。

ComputeResources 屬性

Name (必要, String)

Slurm 佇列運算環境的名稱。名稱最多可有 25 個字元。

更新政策：如果變更此設定，則不允許更新。

InstanceType (必要, String)

此Slurm運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。

叢集組態必須定義 [InstanceType](#) 或 [Instances](#)。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，並想要使用多個子網路，請在 `Instances` 中定義執行個體類型，而不是在 `InstanceType` 中定義。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，或具有多個網路介面或網路介面卡的其他執行個體類型，您必須在私有子網路中啟動運算執行個體，如 [AWS ParallelCluster 使用兩個子網路](#)。AWS 公有 IPs 只能指派給使用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》中的在執行個體[啟動期間指派公有 IPv4 地址](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

Instances (必要)

指定運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](#)。

叢集組態必須定義 [InstanceType](#)或 [Instances](#)。如果兩者都已定義，AWS ParallelCluster 則會失敗。

如需詳細資訊，請參閱[使用 Slurm 進行多個執行個體類型配置](#)。

```
Instances:  
- InstanceType: string
```

Note

從 3.7.0 AWS ParallelCluster 版開始，如果您在[執行個體](#)中設定多個執行個體類型，`EnableMemoryBasedScheduling`則可以啟用。

對於 3.2.0 到 3.6.x AWS ParallelCluster 版，如果您在[執行個體](#)中設定多個執行個體類型，則`EnableMemoryBasedScheduling`無法啟用。

更新政策：對於此清單值設定，可以在更新期間新增新值，或在移除現有值時必須停止運算機群。

InstanceType (必要, String)

要在此Slurm運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，可以是 x86_64 或 arm64。

中列出的執行個體類型[Instances](#)必須具有：

- 相同數量vCPUs，或者，如果 [DisableSimultaneousMultithreading](#) 設定為 true，則為相同數量的核心。
- 相同製造商的相同加速器數量。
- 如果 [Efa Enabled](#) 設定為 true，則支援 EFAtrue。

中列出的執行個體類型[Instances](#)可以有：

- 不同的記憶體數量。

在此情況下，會將最小記憶體設定為消耗性Slurm資源。

Note

從 3.7.0 AWS ParallelCluster 版開始，如果您在[執行個體](#)中設定多個執行個體類型，`EnableMemoryBasedScheduling`則可以啟用。

對於 3.2.0 到 3.6.x AWS ParallelCluster 版，如果您在[執行個體](#)中設定多個執行個體類型，則`EnableMemoryBasedScheduling`無法啟用。

- 不同的網路卡。

在此情況下，為運算資源設定的網路介面數量是由具有最小網路卡數量的執行個體類型所定義。

- 不同的網路頻寬。

- 不同的執行個體存放區大小。

如果您定義 p4d 或 hpc6id 執行個體類型，或具有多個網路介面或網路介面卡的另一個執行個體類型，您必須在私有子網路中啟動運算執行個體，如中所述 [AWS ParallelCluster 使用兩個子網路](#)。AWS 公有 IPs 只能指派給使用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》中的在執行個體 [啟動期間指派公有 IPv4 地址](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

Note

Instances 從 3.3.0 AWS ParallelCluster 版開始支援。

MinCount (選用, Integer)

Slurm 運算資源使用的執行個體數量下限。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](#)

更新政策：必須停止運算機群，才能變更此設定以進行更新。

MaxCount (選用, Integer)

Slurm 運算資源使用的執行個體數目上限。預設為 10。

當您使用時 CapacityType = CAPACITY_BLOCK，MaxCount 必須等於 MinCount 且大於 0，因為容量區塊保留的所有執行個體都是以靜態節點管理。

在叢集建立時間，前端節點會等待所有靜態節點就緒，再發出叢集建立成功的訊號。不過，當您使用時 CapacityType = CAPACITY_BLOCK，與此檢查不會考慮與容量區塊相關聯的運算資源節點部分。即使並非所有設定的容量區塊都作用中，也會建立叢集。

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 (選用, Integer)

佇列運算資源中動態節點的優先順序。優先順序會映射至運算資源動態Slurm節點的節點[Weight](#)組態參數。預設值為 1000。

Slurm 會先排定具有最低Weight值的節點的優先順序。

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任務排程的速率。在 3.7.0 版之前的 AWS ParallelCluster 版本中，靜態和動態節點都獲指派與相同的預設權重。1在此情況下，Slurm可能會因為靜態和動態節點的命名結構描述，而將閒置動態節點優先於閒置靜態節點。當所有其他項目相等時，會依名稱字母順序Slurm排程節點。

Note

DynamicNodePriority 已新增至 3.7.0 AWS ParallelCluster 版。

[更新政策：此設定可以在更新期間變更。](#)

StaticNodePriority (選用, Integer)

佇列運算資源中靜態節點的優先順序。優先順序會映射至運算資源靜態Slurm節點的節點[Weight](#)組態參數。預設值為 1。

Slurm 會先排定具有最低Weight值的節點優先順序。

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任務排程的速率。

Note

StaticNodePriority 已新增至 3.7.0 AWS ParallelCluster 版。

[更新政策：此設定可以在更新期間變更。](#)

SpotPrice (選用, Float)

啟動任何執行個體之前為 Amazon EC2 Spot 執行個體支付的最高價格。預設值為隨需價格。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

DisableSimultaneousMultithreading (選用, Boolean)

如果為 true，則會停用Slurm佇列中節點的多執行緒。預設值為 false。

並非所有執行個體類型都可以停用多執行緒。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》中的[每個執行個體類型的 CPU 核心和每個 CPU 核心的執行緒](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

SchedulableMemory (選用, Integer)

在運算資源之運算節點RealMemory的 Slurm 參數中設定的 MiB 記憶體量。此值是[EnableMemoryBasedScheduling](#)啟用 / [SlurmSettings](#) 時可供任務使用的節點記憶體上限。預設值為 [Amazon EC2 執行個體類型](#)中列出的記憶體的 95%，並由 Amazon EC2 API [DescribeInstanceTypes](#) 傳回。請務必將 GiB 中提供的值轉換為 MiB。

支援的值：1-EC2Memory

EC2Memory 是 [Amazon EC2 執行個體類型](#)中列出的記憶體 (MiB)，並由 Amazon EC2 API [DescribeInstanceTypes](#) 傳回。請務必將 GiB 中提供的值轉換為 MiB。

[EnableMemoryBasedScheduling](#) 啟用 / [SlurmSettings](#) 時，此選項最為相關。如需詳細資訊，請參閱[Slurm 記憶體型排程](#)。

Note

SchedulableMemory 從 AWS ParallelCluster 3.2.0 版開始支援。根據預設，從 3.2.0 版開始，RealMemory 會將Slurm運算節點 AWS ParallelCluster 設定為 Amazon EC2 API 所傳回記憶體的 95%DescribeInstanceTypes。此組態與 的值無關EnableMemoryBasedScheduling。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

HealthChecks (選用)

指定運算資源的運作狀態檢查。

Gpu (選用)

在運算資源上指定 GPU 運作狀態檢查。

Enabled (選用, Boolean)

是否在計算佇列中的資源時 AWS ParallelCluster 執行 GPU 運作狀態檢查。預設值為 `false`。

Note

AWS ParallelCluster 在使用 HealthChecks ARM 作業系統的節點 Gpu 中不支援 `alinux2 /`。這些平台不支援 [NVIDIA 資料中心 GPU Manager \(DCGM\)](#)。

Gpu 運作狀態檢查行為

- 如果 Gpu / Enabled 設定為 `true`，則對運算資源 AWS ParallelCluster 執行運作狀態 GPU 運作狀態檢查。
- Gpu 運作狀態檢查會對運算資源執行運作狀態檢查，以防止在 GPU 降級的節點上提交任務。
- 如果運算節點未通過 Gpu 運作狀態檢查，則運算節點狀態會變更為 DRAIN。新任務不會在此節點上啟動。現有任務會執行至完成。在所有執行中的任務完成後，如果運算節點是動態節點，則運算節點會終止，如果是靜態節點，則會將其取代。
- Gpu 運作狀態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 GPUs 數量，以及 Gpu 運作狀態檢查目標的數量（相當於任務 GPU 目標的數量）。對於具有 8 GPUs 的執行個體，一般持續時間少於 3 分鐘。
- 如果 Gpu 運作狀態檢查在不支援的執行個體上執行，它會結束，並在運算節點上執行任務。例如，如果執行個體沒有 GPU，或者如果執行個體有 GPU，但不是 NVIDIA GPU，則運作狀態檢查會結束，並在運算節點上執行任務。僅支援 NVIDIA GPUs。
- Gpu 運作狀態檢查使用 `dcgmi` 工具在節點上執行運作狀態檢查，並採取下列步驟：

在節點中開始 Gpu 運作狀態檢查時：

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，Gpu 運作狀態檢查會啟動這些服務。
3. 它會偵測持續性模式是否已啟用。
4. 如果未啟用持久性模式，則 Gpu 運作狀態檢查會啟用該模式。

在運作狀態檢查結束時，Gpu 運作狀態檢查會將這些服務和資源還原為其初始狀態。

- 如果任務指派給一組特定的節點 GPUs，則 Gpu 運作狀態檢查只會在該特定集合上執行。否則，Gpu 運作狀態檢查會在節點中的所有 GPUs 上執行。
- 如果運算節點同時收到 2 個或多個 Gpu 運作狀態檢查請求，則只會略過第一個運作狀態檢查和其他項目。這也適用於以節點 GPUs 為目標的運作狀態檢查。您可以檢查日誌檔案，以取得有關此情況的其他資訊。
- `/var/log/parallelcluster/slurm_health_check.log` 檔案中提供了特定運算節點的運作狀態檢查日誌。此檔案可在叢集 Amazon CloudWatch CloudWatch 中取得，您可以在其中找到：
 - Gpu 運作狀態檢查所執行動作的詳細資訊，包括啟用和停用服務和持久性模式。
 - GPU 識別符、序列 ID 和 UUID。
 - 運作狀態檢查輸出。

更新政策：此設定可以在更新期間變更。

Note

HealthChecks 從 3.6.0 AWS ParallelCluster 版開始支援。

Efa (選用)

指定 Slurm 佇列中節點的 Elastic Fabric Adapter (EFA) 設定。

Efa:

Enabled: *boolean*

GdrSupport: *boolean*

更新政策：必須停止運算機群，或 QueueUpdateStrategy 必須設定此設定才能變更更新。

Enabled (選用, Boolean)

指定已啟用 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 Linux 執行個體使用者指南》中的 [支援的執行個體類型](#)。如需詳細資訊，請參閱 [Elastic Fabric Adapter](#)。我們建議您使用叢集 [SlurmQueues / Networking / PlacementGroup](#) 將執行個體之間的延遲降至最低。

預設值為 `false`。

Note

不同可用區域不支援 Elastic Fabric Adapter (EFA)。如需詳細資訊，請參閱 [SubnetIds](#)。

Warning

如果您要在 [SecurityGroups](#) 中定義自訂安全群組，請確定已啟用 EFA 的執行個體是允許所有傳入和傳出流量本身的安全群組成員。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

GdrSupport (選用, **Boolean**)

(選用) 從 3.0.2 AWS ParallelCluster 版開始，此設定沒有效果。如果 Slurm 運算資源和作業系統的執行個體類型支援 GPUDirect RDMA (遠端直接記憶體存取)，則一律會啟用 Elastic Fabric Adapter (EFA) 支援。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

CapacityReservationTarget

[CapacityReservationTarget](#):

[CapacityReservationId](#): *string*

[CapacityReservationResourceGroupArn](#): *string*

指定要用於運算資源的隨需容量保留。

CapacityReservationId (選用, **String**)

佇列運算資源目標的現有容量保留 ID。ID 可以參考 ML 的 [ODCR](#) 或容量區塊。 <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-capacity-blocks.html>

在運算資源層級指定此參數時，InstanceType 是選用的，它會自動從保留中擷取。

CapacityReservationResourceGroupArn (選用, **String**)

指出做為運算資源容量保留之服務連結群組的資源群組的 Amazon Resource Name (ARN)。會從群組 AWS ParallelCluster 識別並使用最適當的容量保留。資源群組必須為運算資源列出的每

個執行個體類型至少有一個 ODCR。如需詳細資訊，請參閱[使用隨需容量預留 \(ODCR\) 啟動執行個體](#)。

- 如果在 [SlurmQueues / Networking](#) 或 [SlurmQueues / ComputeResources](#) / 中啟用 PlacementGroup [Networking](#)，則選取以執行個體類型為目標的資源群組，並在存在時 PlacementGroup 為運算資源 AWS ParallelCluster 選取資源群組。

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 [SlurmQueues / ComputeResources](#) / 中啟用 [Networking](#)，如果存在，AWS ParallelCluster 則選取僅以運算資源的執行個體類型為目標的資源群組。

[更新政策：必須停止運算機群，或 QueueUpdateStrategy 必須設定此設定才能變更更新。](#)

Note

CapacityReservationTarget 已新增 AWS ParallelCluster 3.3.0 版。

Networking

[Networking](#):

[PlacementGroup](#):

[Enabled](#): *boolean*

[Name](#): *string*

[更新政策：所有運算節點都必須停止，才能刪除受管置放群組。必須停止或 QueueUpdateStrategy 設定運算機群，才能變更此設定以進行更新。](#)

PlacementGroup (選用)

指定運算資源的置放群組設定。

Enabled (選用, Boolean)

指出是否將置放群組用於運算資源。

- 如果設定為 true，但未 Name 定義，則無論 [Networking / SlurmQueues PlacementGroup](#) 設定為何，都會指派該運算資源自己的受管置放群組。
- 如果設為 true 且已 Name 定義，則該運算資源會被指派具名置放群組，無論 [SlurmQueues // Networking PlacementGroup](#) 設定為何。

[更新政策：必須停止運算機群，或 QueueUpdateStrategy 必須設定此設定才能變更更新。](#)

Name (選用, String)

用於運算資源之現有叢集置放群組的置放群組名稱。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Note

- 如果未Name同時設定 PlacementGroup / Enabled和 ，其各自的值會預設為 [SlurmQueues / Networking / PlacementGroup](#)設定。
- 使用容量區塊保留時，不應設定置放群組限制，因為即使容量保留有剩餘容量，仍可能因保留之外的置放限制而發生容量不足錯誤。
- ComputeResources / Networking / 已新增 PlacementGroup 3.3.0 AWS ParallelCluster 版。

CustomSlurmSettings (選用, Dict)

(選用) 定義自訂Slurm節點 (運算資源) 組態設定。

指定套用至Slurm節點 (運算資源) 的自訂Slurm組態參數鍵值對字典。

每個單獨的鍵/值對，例如 Param1: Value1，都會以 格式分別新增至Slurm節點組態列的結尾Param1=Value1。

您只能指定 中未拒絕列出的Slurm組態參數CustomSlurmSettings。如需拒絕清單Slurm組態參數的詳細資訊，請參閱 [拒絕列出的Slurm組態參數 CustomSlurmSettings](#)。

AWS ParallelCluster 只會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。您有責任驗證自訂Slurm組態參數。無效的自訂Slurm組態參數可能會導致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Slurm組態參數的詳細資訊，請參閱 Slurm 文件中的 [slurm.conf](#)。

更新政策：此設定可以在更新期間變更。

Note

CustomSlurmSettings 從 3.6.0 AWS ParallelCluster 版開始支援。

Tags (選用, 【字串】)

標籤鍵值對的清單。ComputeResource標籤會覆寫 [Tags 區段](#) 或 [SlurmQueues](#) / 中指定的重複標籤Tags。

Key (選用, String)

標籤金鑰。

Value (選用, String)

標籤值。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

LaunchTemplateOverrides (選用)

Note

LaunchTemplateOverrides 已新增 AWS ParallelCluster 3.15.0 版。

指定啟動範本以覆寫為運算資源 AWS ParallelCluster 建立的預設啟動範本。啟動範本應該只包含網路介面 overrides. AWS ParallelCluster validation the launch template，並防止覆寫其他參數。如需如何使用此覆寫的詳細資訊，請參閱 [使用啟動範本覆寫自訂運算節點網路介面](#)。

```
LaunchTemplateOverrides:  
  LaunchTemplateId: string  
  Version: string
```

LaunchTemplateId (必要, String)

啟動範本的 ID。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Version (必要, String)

啟動範本的版本編號。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

ComputeSettings

(必要) 定義Slurm佇列的ComputeSettings組態。

ComputeSettings 屬性

指定Slurm佇列中節點ComputeSettings的屬性。

```
ComputeSettings:  
  LocalStorage:  
    RootVolume:  
      Size: integer  
      Encrypted: boolean  
      VolumeType: string  
      Iops: integer  
      Throughput: integer  
    EphemeralVolume:  
      MountDir: string
```

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

LocalStorage (選用)

指定Slurm佇列中節點LocalStorage的屬性。

```
LocalStorage:  
  RootVolume:  
    Size: integer  
    Encrypted: boolean  
    VolumeType: string  
    Iops: integer  
    Throughput: integer  
  EphemeralVolume:  
    MountDir: string
```

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

RootVolume (選用)

指定Slurm佇列中節點根磁碟區的詳細資訊。

```
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  VolumeType: string  
  Iops: integer  
  Throughput: integer
```

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Size (選用, Integer)

指定Slurm佇列中節點的根磁碟區大小，以 GB (GiB) 為單位。預設大小來自 AMI。使用不同的大小需要 AMI 支援 growroot。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Encrypted (選用, Boolean)

如果為 true，Slurm佇列中節點的根磁碟區會加密。預設值為 true。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

VolumeType (選用, String)

指定Slurm佇列中節點的 [Amazon EBS 磁碟區類型](#)。支援的值為 gp2、gp3、io1、io2、st1、sc1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](#)。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Iops (選用, Boolean)

定義 io1、io2和gp3類型磁碟區的 IOPS 數目。

預設值、支援的值和volume_iopsvolume_size比率因 VolumeType和 而異Size。

VolumeType = io1

預設值 Iops = 100

支援的值 Iops = 100–64000 †

最大volume_iopsvolume_size比率 = 每 GiB 50 IOPS。5000 IOPS 需要至少 100 GiB volume_size的。

VolumeType = io2

預設值 Iops = 100

支援的值 Iops = 100–64000 (io2區塊快速磁碟區為 256000) †

最大IopsSize比率 = 每 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB Size的。

VolumeType = gp3

預設 Iops = 3000

支援的值 Iops = 3000–16000 †

對於 IOPS 大於 3000 的磁碟區，最大IopsSize比率 = 每 GiB 500 IOPS。

† 只有建置在 Nitro 系統上且佈建超過 32,000 IOPS 的執行個體，才能保證 IOPS 上限。其他執行個體最多可有 32,000 IOPS。舊版磁碟io1區可能無法達到完整效能，除非您修改磁碟區。io2Block Express 磁碟區在R5b執行個體類型上最多支援 256000 個volume_iops值。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [io2 Block Express 磁碟區](#)。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Throughput (選用, Integer)

定義gp3磁碟區類型的輸送量，以 MiB/s 為單位。此設定僅在 VolumeType為 時有效gp3。預設值為 125。支援的值：125–1000 MiB/s

Throughput 與 的比率Iops不可超過 0.25。1000 MiB/s 的最大輸送量要求Iops設定至少為 4000。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

EphemeralVolume (選用, Boolean)

指定暫時性磁碟區的設定。暫時性磁碟區是透過將所有執行個體存放區磁碟區合併為與ext4檔案系統格式化的單一邏輯磁碟區來建立。預設值為 /scratch。如果執行個體類型沒有任何執行個體存放磁碟區，則不會建立暫時性磁碟區。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*

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

MountDir (選用, String)

Slurm 佇列中每個節點的暫時性磁碟區的掛載目錄。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

CustomActions

(選用) 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
```

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

CustomActions 屬性

OnNodeStart (選用, String)

在啟動任何節點部署引導動作之前，指定要在Slurm佇列中節點上執行的一系列指令碼或單一指令碼。AWS ParallelCluster 不支援同時包含單一指令碼和Sequence相同的自訂動作。如需詳細資訊，請參閱[自訂引導操作](#)。

Sequence (選用)

要執行的指令碼清單。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Script (必要, String)

要使用的檔案。檔案路徑的開頭可以是 https://或 s3://。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Args (選用, [String])

要傳遞至指令碼的引數清單。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

Script (必要, String)

用於單一指令碼的檔案。檔案路徑的開頭可以是 https://或 s3://。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

Args (選用, [String])

要傳遞至單一指令碼的引數清單。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

OnNodeConfigured (選用, String)

指定在所有節點引導動作完成後，要在Slurm佇列中節點上執行的一系列指令碼或單一指令碼。AWS ParallelCluster 不支援同時包含單一指令碼和Sequence相同的自訂動作。如需詳細資訊，請參閱[自訂引導操作](#)。

Sequence (選用)

要執行的指令碼清單。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

Script (必要, String)

要使用的檔案。檔案路徑的開頭可以是 https://或 s3://。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

Args (選用, [String])

要傳遞至指令碼的引數清單。

[更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。](#)

Script (必要, String)

用於單一指令碼的檔案。檔案路徑的開頭可以是 https://或 s3://。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Args (選用, [String])

要傳遞至單一指令碼的引數清單。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

Note

Sequence 從 3.6.0 AWS ParallelCluster 版開始新增。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，而不包含 Sequence。

AWS ParallelCluster 不支援同時包含單一指令碼和Sequence相同的自訂動作。

Iam

(選用) 定義Slurm佇列的選用 IAM 設定。

Iam:

S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

AdditionalIamPolicies:

- Policy: *string*

InstanceProfile: *string*

InstanceRole: *string*

更新政策：此設定可以在更新期間變更。

Iam 屬性

InstanceProfile (選用, String)

指定執行個體描述檔，以覆寫Slurm佇列的預設執行個體角色或執行個體描述檔。您無法同時指定 InstanceProfile和 InstanceRole。格式是 `arn:${Partition}:iam::${Account}:instance-profile/${InstanceProfileName}`。

如果指定此選項，則無法指定 S3Access 和 AdditionalIamPolicies 設定。

我們建議您指定一個或兩個 S3Access 和 AdditionalIamPolicies 設定，因為新增至的功能 AWS ParallelCluster 通常需要新的許可。

[更新政策：必須停止運算機群，才能變更此設定以進行更新。](#)

InstanceRole (選用, 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描述檔。您無法同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:${Partition}:iam::${Account}:role/${RoleName}`。

如果指定此選項，則無法指定 S3Access 和 AdditionalIamPolicies 設定。

我們建議您指定一個或兩個 S3Access 和 AdditionalIamPolicies 設定，因為新增至的功能 AWS ParallelCluster 通常需要新的許可。

[更新政策：此設定可以在更新期間變更。](#)

S3Access (選用)

指定 Slurm 佇列的儲存貯體。這用於產生政策，以授予 Slurm 佇列中儲存貯體的指定存取權。

如果指定此選項，則無法指定 InstanceProfile 和 InstanceRole 設定。

我們建議您指定一個或兩個 S3Access 和 AdditionalIamPolicies 設定，因為新增至的功能 AWS ParallelCluster 通常需要新的許可。

S3Access:

- `BucketName`: *string*
- `EnableWriteAccess`: *boolean*
- `KeyName`: *string*

[更新政策：此設定可以在更新期間變更。](#)

BucketName (必要, String)

儲存貯體的名稱。

[更新政策：此設定可以在更新期間變更。](#)

KeyName (選用, String)

儲存貯體的金鑰。預設值為 `*`。

更新政策：此設定可以在更新期間變更。

EnableWriteAccess (選用, **Boolean**)

指出是否已啟用儲存貯體的寫入存取。

更新政策：此設定可以在更新期間變更。

AdditionalIamPolicies (選用)

指定 Amazon EC2 的 IAM 政策的 Amazon Resource Name (ARNs) 清單。除了 所需的許可之外，此清單還會連接到用於Slurm佇列的根角色 AWS ParallelCluster。

IAM 政策名稱及其 ARN 不同。無法使用名稱。

如果指定此選項，則無法指定 InstanceProfile和 InstanceRole設定。

我們建議您使用 `AdditionalIamPolicies` 因為 `AdditionalIamPolicies` 已新增至 AWS ParallelCluster 所需的許可，且 InstanceRole 必須包含所有必要的許可。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*

更新政策：此設定可以在更新期間變更。

Policy (必要, **[String]**)

IAM 政策清單。

更新政策：此設定可以在更新期間變更。

SlurmSettings

(選用) 定義適用於整個叢集Slurm的 設定。

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 屬性

ScaledownIdleTime (選用, Integer)

定義沒有任務且Slurm節點終止的時間量 (以分鐘為單位)。

預設值為 10。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

MungeKeySecretArn (選用, String)

純文字 AWS Secrets Manager 秘密的 Amazon Resource Name (ARN)，其中包含要在Slurm叢集中使用的 base64 編碼 Munge 金鑰。此 Munge 金鑰將用於驗證Slurm用戶端命令與做為遠端伺服器的協助程式之間的 RPC Slurm 呼叫。如果未提供 MungeKeySecretArn，AWS ParallelCluster 將為叢集產生隨機 Munge 金鑰。

Note

MungeKeySecretArn 從 AWS ParallelCluster 3.8.0 版開始支援。

Warning

如果 MungeKeySecretArn 是新新增至現有叢集，則在復原或稍後移除 MungeKeySecretArn 時，ParallelCluster 不會還原先前的 munge 金鑰。MungeKeySecretArn 反之，將產生新的隨機 munge 金鑰。

如果 AWS ParallelCluster 使用者具有該特定秘密資源的 [DescribeSecret](#) 許可，則會驗證 MungeKeySecretArn。MungeKeySecretArn 在下列情況下有效：

- 指定的秘密存在，且
- 秘密為純文字，並包含有效的 base64 編碼字串，以及
- 解碼的二進位 munge 金鑰的大小介於 256 到 8192 位元之間。

如果叢集使用者 IAM 政策不包含 DescribeSecret，則不會驗證 MungeKeySecretArn，並顯示警告訊息。如需詳細資訊，請參閱[基本 AWS ParallelCluster pcluster 使用者政策](#)。

當您更新 MungeKeySecretArn 時，必須停止運算機群和所有登入節點。

如果在 ARN 保持不變時修改秘密 ARN 中的秘密值，則叢集不會自動使用新的 munge 金鑰更新。若要使用秘密 ARN 的新 Munge 金鑰，您必須停止運算機群和登入節點，然後從前端節點執行下列命令。

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

執行命令後，您可以繼續運算機群和登入節點：新佈建的運算和登入節點會自動開始使用新的 munge 金鑰。

若要產生 base64 編碼的自訂 munge 金鑰，您可以使用隨 [munge 軟體分佈的 mungekey 公用程式](#)，然後使用作業系統中一般提供的 base64 公用程式進行編碼。或者，您可以使用 bash（請將 bs 參數設定為 32 到 1024）

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

或 Python，如下所示：

```
import random
import os
import base64

# key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

[更新政策：必須停止運算機群和登入節點，才能變更此設定以進行更新。](#)

QueueUpdateStrategy (選用, String)

指定具有下列更新政策之 [SlurmQueues](#) 區段參數的取代策略：

更新政策：必須停止運算機群，或QueueUpdateStrategy必須設定此設定才能變更更新。

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

具有變更數值的佇列中的節點會設定為 DRAINING。處於此狀態的節點不接受新任務，且執行中的任務會繼續完成。

在節點變成 idle(DRAINED) 之後，如果節點是靜態節點，則會取代節點，如果節點是動態節點，則會終止節點。其他佇列中沒有變更數值的其他節點不會受到影響。

此策略需要以變更的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

QueueUpdateStrategy 數值的預設值。使用此設定，更新 [SlurmQueues](#) 區段下的數值需要您在執行叢集更新之前[停止運算機群](#)：

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

TERMINATE

在數值變更的佇列中，執行中的任務會終止，且節點會立即關閉電源。

取代靜態節點，並終止動態節點。

其他佇列中沒有變更數值的其他節點不會受到影響。

更新政策：更新期間不會分析此設定。

Note

QueueUpdateStrategy 從 AWS ParallelCluster 3.2.0 版開始支援。

EnableMemoryBasedScheduling (選用，Boolean)

如果為 true，則會在 中啟用記憶體型排程Slurm。如需詳細資訊，請參閱 [SlurmQueues](#) / [ComputeResources](#) / [SchedulableMemory](#)。

預設值為 `false`。

Warning

啟用記憶體型排程會影響Slurm排程器處理任務和節點配置的方式。
如需詳細資訊，請參閱[Slurm 記憶體型排程](#)。

Note

`EnableMemoryBasedScheduling` 從 3.2.0 AWS ParallelCluster 版開始支援。

Note

從 3.7.0 AWS ParallelCluster 版開始，如果您在[執行個體](#)中設定多個執行個體類型，`EnableMemoryBasedScheduling`則可以啟用。
對於 3.2.0 到 3.6.x AWS ParallelCluster 版，如果您在[執行個體](#)中設定多個執行個體類型，則`EnableMemoryBasedScheduling`無法啟用。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

CustomSlurmSettings (選用, [Dict])

定義套用至整個叢集的自訂Slurm設定。

指定要附加至 AWS ParallelCluster 產生之`slurm.conf`檔案結尾的鍵值對Slurm組態字典清單。

清單中的每個字典都會顯示為新增至Slurm組態檔案的個別行。您可以指定簡單或複雜的參數。

簡單參數由單一金鑰對組成，如下列範例所示：

```
- Param1: 100  
- Param2: "SubParam1,SubParam2=SubValue2"
```

Slurm 組態中呈現的範例：

```
Param1=100  
Param2=SubParam1,SubParam2=SubValue2
```

複雜的Slurm組態參數由多個空格分隔的索引鍵值組成，配對如下列範例所示：

```
- 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，以Slurm組態轉譯：

```
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 AWS ParallelCluster。

如果您在 `CustomSlurmSettings` 中指定自訂Slurm組態參數 `CustomSlurmSettings`，則不得為 `CustomSlurmSettingsIncludeFile` 指定自訂Slurm組態參數。

您只能指定 `CustomSlurmSettings` 中未拒絕列出的Slurm組態參數。如需拒絕列出Slurm組態參數的詳細資訊，請參閱 [拒絕列出的Slurm組態參數 CustomSlurmSettings](#)。

AWS ParallelCluster 只會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。您有責任驗證自訂Slurm組態參數。無效的自訂Slurm組態參數可能會導致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 `CustomSlurmSettings` 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Slurm組態參數的詳細資訊，請參閱 Slurm 文件中的 [slurm.conf](#)。

[更新政策：此設定可以在更新期間變更。](#)

Note

CustomSlurmSettings 從 AWS ParallelCluster 3.6.0 版開始支援。

CustomSlurmSettingsIncludeFile (選用, **String**)

定義套用至整個叢集的自訂Slurm設定。

指定要在 AWS ParallelCluster 產生的Slurm檔案結尾附加的自訂Slurm組態參數組成的自訂slurm.conf檔案。

您必須包含 檔案的路徑。路徑的開頭可以是 https://或 s3://。

如果您為 指定自訂Slurm組態參數CustomSlurmSettingsIncludeFile, 則不得為 指定自訂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 AWS ParallelCluster。

您只能指定 中未拒絕列出的Slurm組態參數CustomSlurmSettingsIncludeFile。如需拒絕清單Slurm組態參數的詳細資訊, 請參閱 [拒絕列出的Slurm組態參數 CustomSlurmSettings](#)。

AWS ParallelCluster 只會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。您有責任驗證自訂Slurm組態參數。無效的自訂Slurm組態參數可能會導致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Slurm組態參數的詳細資訊, 請參閱 Slurm 文件中的 [slurm.conf](#)。

[更新政策](#): 此設定可以在更新期間變更。

Note

CustomSlurmSettings 從 AWS ParallelCluster 3.6.0 版開始支援。

Database

(選用) 定義在叢集上啟用Slurm會計的設定。如需詳細資訊，請參閱[Slurm 使用 會計 AWS ParallelCluster](#)。

Database:

`Uri`: *string*

`UserName`: *string*

`PasswordSecretArn`: *string*

更新政策：必須停止運算機群，才能變更此設定以進行更新。

Database 屬性

Uri (必要, String)

做為Slurm會計後端的資料庫伺服器地址。此 URI 必須格式化為 `host:port`且不得包含結構描述，例如 `mysql://`。主機可以是 IP 地址或可由前端節點解析的 DNS 名稱。如果未提供連接埠，AWS ParallelCluster 會使用MySQL預設連接埠 3306。

AWS ParallelCluster 會將Slurm會計資料庫引導至叢集，且必須存取資料庫。

發生下列情況之前，資料庫必須能夠連線：

- 叢集已建立。
- Slurm 已透過叢集更新啟用 會計。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

UserName (必要, String)

Slurm 用來連線至資料庫、撰寫會計日誌和執行查詢的身分。使用者必須同時擁有資料庫的讀取和寫入許可。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

PasswordSecretArn (必要, String)

包含UserName純文字密碼之 AWS Secrets Manager 秘密的 Amazon Resource Name (ARN)。此密碼會與 UserName 和 Slurm 會計一起使用，以在資料庫伺服器上進行身分驗證。

Note

- 當您使用 AWS Secrets Manager 主控台建立秘密時，請務必選取「其他秘密類型」、選取純文字，並只在秘密中包含密碼文字。
- 您無法在資料庫密碼中使用 '#' 字元，因為 Slurm 在 slurmdbd.conf 中不支援它。
- 如需如何使用 AWS Secrets Manager 建立秘密的詳細資訊，請參閱[建立 AWS Secrets Manager 秘密](#)。

如果使用者具有 [DescribeSecret](#) 的許可，PasswordSecretArn 則會驗證。如果指定的秘密存在，PasswordSecretArn 則有效。如果使用者 IAM 政策不包含 DescribeSecret，PasswordSecretArn 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 使用者政策](#)。

當您更新 PasswordSecretArn，必須停止運算機群。如果秘密值變更，且秘密 ARN 未變更，則叢集不會自動更新為新的資料庫密碼。若要更新叢集的新秘密值，您必須在運算機群停止後，從前端節點內執行下列命令。

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

Warning

我們建議您只在運算機群停止時變更資料庫密碼，以避免遺失會計資料。

[更新政策：必須停止運算機群，才能變更此設定以進行更新。](#)

DatabaseName (選用, String)

要用於 Slurm 會計之資料庫伺服器上的資料庫名稱（由 Uri 參數定義）。

資料庫的名稱可能包含小寫字母、數字和底線。名稱長度不可超過 64 個字元。

此參數映射至 [slurmdbd.conf](#) 的 StorageLoc 參數。

如果 DatabaseName 未提供，則 ParallelCluster 將使用叢集的名稱來定義 StorageLoc 的值。

DatabaseName 允許更新，但有下列考量：

- 如果資料庫伺服器上尚不存在名為 DatabaseName 的資料庫，則 slurmdbd 會建立該資料庫。您有責任視需要重新設定新資料庫（例如新增會計實體 - 叢集、帳戶、使用者、關聯、QOSs 等）。
- 如果資料庫伺服器上已存在名為 DatabaseName 的資料庫，slurmdbd 會將其用於Slurm會計功能。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

Note

Database 從 3.3.0 版開始新增。

ExternalSlurmdbd

(選用) 定義使用外部 slurmdbd 伺服器啟用Slurm會計的設定。如需詳細資訊，請參閱[Slurm使用進行會計 AWS ParallelCluster](#)。

ExternalSlurmdbd:

Host: *string*

Port: *integer*

ExternalSlurmdbd 屬性

Host (必要, String)

外部 slurmdbd 伺服器用於Slurm會計的地址。主機可以是 IP 地址或可由前端節點解析的 DNS 名稱。

更新政策：此設定可以在更新期間變更。

Port (選用, Integer)

slurmdbd 服務接聽的連接埠。預設值為 6819。

更新政策：此設定可以在更新期間變更。

Dns

(選用) 定義適用於整個叢集Slurm的設定。

Dns:

`DisableManagedDns`: *boolean*

`HostedZoneId`: *string*

`UseEc2Hostnames`: *boolean*

Dns 屬性**DisableManagedDns (選用, Boolean)**

如果 `true`，則不會建立叢集的 DNS 項目，也無法解析 Slurm 節點名稱。

根據預設，AWS ParallelCluster 會建立 Route 53 託管區域，其中節點會在啟動時註冊。預設值為 `false`。如果 `DisableManagedDns` 設定為 `true`，則不會建立託管區域 AWS ParallelCluster。

若要了解如何使用此設定在無法存取網際網路的子網路中部署叢集，請參閱 [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#)。

⚠ Warning

叢集需要名稱解析系統才能正常運作。如果 `DisableManagedDns` 設定為 `true`，您必須提供名稱解析系統。若要使用 Amazon EC2 預設 DNS，請將 `UseEc2Hostnames` 設定為 `true`。或者，設定您自己的 DNS 解析程式，並確保在執行個體啟動時註冊節點名稱。例如，您可以透過設定 [CustomActions](#) / 來執行此操作 [OnNodeStart](#)。

更新政策：如果變更此設定，則不允許更新。

HostedZoneId (選用, String)

定義要用於叢集 DNS 名稱解析的自訂 Route 53 託管區域 ID。提供時，會在指定的託管區域中 AWS ParallelCluster 註冊叢集節點，而不會建立受管託管區域。

更新政策：如果變更此設定，則不允許更新。

UseEc2Hostnames (選用, Boolean)

如果為 `true`，叢集運算節點會使用預設的 EC2 主機名稱設定。Slurm NodeHostName 也會更新此資訊。預設值為 `false`。

若要了解如何使用此設定在無法存取網際網路的子網路中部署叢集，請參閱 [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#)。

Note

此備註從 AWS ParallelCluster 3.3.0 版開始並不相關。

對於 3.3.0 之前的 AWS ParallelCluster 支援版本：

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

- `prolog` 會執行 以在每個任務配置時，將節點資訊新增至運算節點 `/etc/hosts` 上的。
- `epilog` 會執行 以清除 撰寫的內容 `prolog`。

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

更新政策：如果變更此設定，則不允許更新。

SharedStorage 區段

(選用) 叢集的共用儲存設定。

AWS ParallelCluster 支援使用 [Amazon EBS](#)、[FSx for ONTAP](#) 和 [FSx for OpenZFS](#) 共用儲存磁碟區、[Amazon EFS](#) 和 [FSx for Lustre](#) 共用儲存檔案系統，或 [檔案快取](#)。

在 `SharedStorage` 區段中，您可以定義外部或受管儲存：

- 外部儲存是指您管理的現有磁碟區或檔案系統。AWS ParallelCluster 不會建立或刪除它。
- AWS ParallelCluster 受管儲存是指 AWS ParallelCluster 已建立且可刪除的磁碟區或檔案系統。

如需 [共用儲存配額](#) 和如何設定共用儲存的詳細資訊，請參閱使用 AWS ParallelCluster [共用儲存](#) 中的。

Note

如果 AWS Batch 用作排程器，則 FSx for Lustre 僅適用於叢集前端節點。

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
  AccessPointId: 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/外部 EBS、受管 EFS 和受管 FSx Lustre，更新政策為 [更新政策：對於此清單值設定，運算機群必須停止或QueueUpdateStrategy必須設定為新增值；在移除現有值時，運算機群必須停止。](#)
- 對於外部 EFS、FSx Lustre、FSx ONTAP、FSx OpenZfs 和檔案快取，更新政策為：[更新政策：此設定可以在更新期間變更。](#)

SharedStorage 屬性

MountDir (必要, String)

掛載共用儲存體的路徑。

更新政策：如果變更此設定，則不允許更新。

Name (必要, String)

共用儲存體的名稱。您在更新設定時使用此名稱。

Warning

如果您指定 AWS ParallelCluster 受管共用儲存體，並變更的值Name，則會刪除現有的受管共用儲存體和資料，並建立新的受管共用儲存體。Name 使用叢集更新變更的值等同於將現有的受管共用儲存取代為新的儲存。Name 如果您需要保留現有共用儲存體中的資料，請務必在變更之前備份資料。

更新政策：對於此清單值設定，運算機群必須停止或QueueUpdateStrategy必須設定為新增值；在移除現有值時，運算機群必須停止。

StorageType (必要, String)

共用儲存體的類型。支援的值為 Ebs、Efs、FsxOntap、FsxLustre和 FsxOpenZfs。

如需詳細資訊，請參

閱[FsxLustreSettings](#)、[FsxOntapSettings](#)及[FsxOpenZfsSettings](#)。

Note

如果您使用 AWS Batch 做為排程器，FSx for Lustre 只能在叢集前端節點上使用。

更新政策：如果變更此設定，則不允許更新。

EbsSettings

(選用) 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
```

更新政策：如果變更此設定，則不允許更新。

EbsSettings 屬性

當 [DeletionPolicy](#) 設定為 `Delete`，如果刪除叢集或使用叢集更新移除磁碟區，則會刪除具有其資料的受管磁碟區。

如需詳細資訊，請參閱《使用》[共用儲存](#)中的。AWS ParallelCluster

VolumeType (選用, String)

指定 [Amazon EBS 磁碟區類型](#)。支援的值為 `gp2`、`gp3`、`io1`、`io2`、`st1`、`sc1`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](#)。

更新政策：如果變更此設定，則不允許更新。

Iops (選用, Integer)

定義 `io1`、`io2`和`gp3`類型磁碟區的 IOPS 數目。

預設值、支援的值和`volume_iopsvolume_size`比率因 VolumeType和 而異Size。

VolumeType = `io1`

預設值 Iops = 100

支援的值 Iops = 100–64000 †

最大`volume_iopsvolume_size`比率 = 每個 GiB 50 IOPS。5000 IOPS 需要至少 100 GiB `volume_size`的。

VolumeType = `io2`

預設值 Iops = 100

支援的值 Iops = 100–64000 (io2 區塊快速磁碟區為 256000) †

最大IopsSize比率 = 每個 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB Size 的。

VolumeType = gp3

預設 Iops = 3000

支援的值 Iops = 3000–16000

最大IopsSize比率 = 每個 GiB 500 IOPS。5000 IOPS 需要至少 10 GiB Size 的。

† 只有在佈建超過 32,000 IOPS 的 [Nitro 系統上建置的執行個體上](#)，才能保證 IOPS 上限。其他執行個體保證高達 32,000 IOPS。除非您 [修改磁碟區](#)，否則舊版磁碟 io1 區可能無法達到完整效能。io2Block Express 磁碟區在 R5b 執行個體類型上最多支援 256000 個 volume_iops 值。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [io2 Block Express 磁碟區](#)。

[更新政策：此設定可以在更新期間變更。](#)

Size (選用, Integer)

以 GB (GiB) 為單位指定磁碟區大小。預設值為 35。

[更新政策：如果變更此設定，則不允許更新。](#)

Encrypted (選用, Boolean)

指定磁碟區是否已加密。預設值為 true。

[更新政策：如果變更此設定，則不允許更新。](#)

KmsKeyId (選用, String)

指定用於加密的自訂 AWS KMS 金鑰。此設定需要將 Encrypted 設定設為 true。

[更新政策：如果變更此設定，則不允許更新。](#)

SnapshotId (選用, String)

如果您使用快照做為磁碟區的來源，請指定 Amazon EBS 快照 ID。

[更新政策：如果變更此設定，則不允許更新。](#)

VolumeId (選用, String)

指定 Amazon EBS 磁碟區 ID。為 EbsSettings 執行個體指定此值時，也只能指定 MountDir 參數。

磁碟區必須在與 相同的可用區域中建立HeadNode。

Note

3.4.0 AWS ParallelCluster 版中新增了多個可用區域。

更新政策：如果變更此設定，則不允許更新。

Throughput (選用, Integer)

為磁碟區佈建的輸送量，以 MiB/s 為單位，上限為 1,000 MiB/s。

此設定僅在 VolumeType 為 時有效gp3。支援的範圍為 125 到 1000，預設值為 125。

更新政策：此設定可以在更新期間變更。

DeletionPolicy (選用, String)

指定在刪除叢集或移除磁碟區時，是否應保留、刪除或快照磁碟區。支援的值為 Delete、Retain和 Snapshot。預設值為 Delete。

將 [DeletionPolicy](#) 設定為 時Delete，如果刪除叢集，或使用叢集更新移除磁碟區，則會刪除具有其資料的受管磁碟區。

如需詳細資訊，請參閱[共用儲存](#)。

更新政策：此設定可以在更新期間變更。

Note

DeletionPolicy 從 3.2.0 AWS ParallelCluster 版開始支援。

Raid

(選用) 定義 RAID 磁碟區的組態。

Raid:

Type: *string*

NumberOfVolumes: *integer*

更新政策：如果變更此設定，則不允許更新。

Raid 屬性

Type (必要, String)

定義 RAID 陣列的類型。支援的值為 "0" (條紋) 和 "1" (鏡像)。

更新政策：如果變更此設定，則不允許更新。

NumberOfVolumes (選用, Integer)

定義用於建立 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。

更新政策：如果變更此設定，則不允許更新。

EfsSettings

(選用) 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*

AccessPointId: *string*

更新政策：如果變更此設定，則不允許更新。

EfsSettings 屬性

當 DeletionPolicy 設定為 `Delete`，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除包含其資料的受管檔案系統。

如需詳細資訊，請參閱《使用》[共用儲存](#)中的。AWS ParallelCluster

Encrypted (選用, Boolean)

指定 Amazon EFS 檔案系統是否已加密。預設值為 `false`。

更新政策：如果變更此設定，則不允許更新。

KmsKeyId (選用, String)

指定用於加密的自訂 AWS KMS 金鑰。此設定需要將 Encrypted 設定設為 true。

更新政策：如果變更此設定，則不允許更新。

EncryptionInTransit (選用, Boolean)

如果設定為 true，則會使用 Transport Layer Security (TLS) 掛載 Amazon EFS 檔案系統。根據預設，這會設定為 false。

Note

如果 AWS Batch 用作排程器，EncryptionInTransit 則不支援。

Note

EncryptionInTransit 從 3.4.0 版開始 AWS ParallelCluster 新增。

更新政策：如果變更此設定，則不允許更新。

IamAuthorization (選用, Boolean)

IamAuthorization 從 3.4.0 AWS ParallelCluster 版開始新增。

如果設定為 true，Amazon EFS 會使用系統的 IAM 身分進行身分驗證。根據預設，這會設定為 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，IamAuthorization 則不支援。

更新政策：如果變更此設定，則不允許更新。

PerformanceMode (選用, 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。不過，大多數檔案操作的延遲會略高。

更新政策：如果變更此設定，則不允許更新。

ThroughputMode (選用, 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`。使用 `provisioned` 時，`ProvisionedThroughput` 必須指定。

更新政策：此設定可以在更新期間變更。

ProvisionedThroughput (當 ThroughputMode 為 `provisioned`、時必填 Integer)

定義 Amazon EFS 檔案系統的佈建輸送量（以 MiB/秒為單位），以 MiB/秒為單位。這對應至 Amazon EFS API 參考中的 [ProvisionedThroughputInMibps](#) 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。聯絡支援以請求增加限制。

更新政策：此設定可以在更新期間變更。

FileSystemId (選用, 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。


指定此選項時，`MountDir` 只能指定。EfsSettings 無法指定其他。

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現有的掛載目標，允許來自 `HeadNode` 的傳入和傳出 NFS 流量 `ComputeNodes`。在 [排程/SlurmQueues](#) / [聯網/SubnetIds](#) `SubnetIds` 中設定多個可用區域。


若要確保叢集和檔案系統之間允許流量，您可以執行下列其中一項操作：

- 設定掛載目標的安全群組，以允許進出叢集子網路 CIDR 或字首清單的流量。


 Note

AWS ParallelCluster 會驗證連接埠是否開啟，以及 CIDR 或字首清單是否已設定。AWS ParallelCluster 不會驗證 CIDR 區塊或字首清單的內容。


- 使用 // [Networking SecurityGroups](#) 和 [SlurmQueues / HeadNode / Networking](#) 為叢集節點設定自訂安全群組 [SecurityGroups](#)。自訂安全群組必須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 只會驗證連接埠是否開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

 Warning

只有在所有運算節點和前端節點都位於相同的可用區域時，才支援 EFS OneZone。EFS OneZone 只能有一個掛載目標。

 Note

3.4.0 AWS ParallelCluster 版中新增了多個可用區域。

[更新政策：如果變更此設定，則不允許更新。](#)

DeletionPolicy (選用, String)

指定從叢集移除檔案系統或刪除叢集時，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 [DeletionPolicy](#) 設定為 Delete 時，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除包含其資料的受管檔案系統。

如需詳細資訊，請參閱[共用儲存](#)。

更新政策：此設定可以在更新期間變更。

Note

DeletionPolicy 從 3.3.0 AWS ParallelCluster 版開始支援。

AccessPointId (選用, String)

如果指定此選項，access point ID則會掛載定義的檔案系統進入點，而不是檔案系統根目錄。

如需詳細資訊，請參閱[共用儲存](#)。

更新政策：如果變更此設定，則不允許更新。

FsxLustreSettings

Note

您必須定義 FsxLustreSettings FsxLustre 是否為指定 [StorageType](#)。

(選用) FSx for Lustre 檔案系統的設定。

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
```

更新政策：如果變更此設定，則不允許更新。

Note

如果 AWS Batch 用作排程器，則 FSx for Lustre 僅適用於叢集前端節點。

FsxLustreSettings 屬性

當 [DeletionPolicy](#) 設定為 `Delete`，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除包含其資料的受管檔案系統。

如需詳細資訊，請參閱[共用儲存](#)。

StorageCapacity (必要, Integer)

設定 FSx for Lustre 檔案系統的儲存容量，以 GiB 為單位。如果您要建立新的檔案系統，StorageCapacity 則需要。StorageCapacity 如果指定 `FileSystemId` `BackupId` 或 `FileSystemId`，請勿包含。

- 對於 `SCRATCH_2`、`PERSISTENT_1` 和 `PERSISTENT_2` 部署類型，有效值為 1200 GiB、2400 GiB，增量為 2400 GiB。
- 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。

更新政策：如果變更此設定，則不允許更新。

DeploymentType (選用, String)

指定 FSx for Lustre 檔案系統的部署類型。支援的值為 `SCRATCH_1`、`SCRATCH_2`、`PERSISTENT_1` 和 `PERSISTENT_2`。預設值為 `SCRATCH_2`。

當您需要暫時儲存 `SCRATCH_1` 和較短時間處理資料時，請選擇 `SCRATCH_2` 部署類型。`SCRATCH_2` 部署類型提供資料傳輸中加密，以及高於 `SCRATCH_1` 的爆量輸送量容量。

選擇長期儲存的 `PERSISTENT_1` 部署類型，以及非延遲敏感的輸送量導向工作負載。`PERSISTENT_1` 支援傳輸中資料的加密。它可用於 FSx for Lustre 提供的所有 AWS 區域。

針對需要最高 IOPS 和輸送量層級的延遲敏感工作負載，選擇 PERSISTENT_2 部署類型。PERSISTENT_2 支援 SSD 儲存並提供更高的 PerUnitStorageThroughput (最高 1000 MB/s/TiB)。PERSISTENT_2 提供有限數量的 AWS 區域。如需部署類型和 PERSISTENT_2 可用 AWS 區域清單的詳細資訊，請參閱《Amazon FSx for Lustre 使用者指南》中的 [FSx for Lustre 的檔案系統部署選項](#)。FSx

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，會自動啟用傳輸中的資料加密。

從支援的 [中](#) 支援的執行個體類型存取時 PERSISTENT_1，支援加密 SCRATCH_2、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

3.2.0 AWS ParallelCluster 版已新增對 PERSISTENT_2 部署類型的支援。

[更新政策：如果變更此設定，則不允許更新。](#)

ImportedFileChunkSize (選用, Integer)

對於從資料儲存庫匯入的檔案，此值會決定存放在單一實體磁碟上每個檔案（以 MiB 為單位）的條紋計數和最大資料量。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](#)。

[更新政策：如果變更此設定，則不允許更新。](#)

DataCompressionType (選用, String)

設定 FSx for Lustre 檔案系統的資料壓縮組態。支援的值為 LZ4。LZ4 表示使用 LZ4 演算法開啟資料壓縮。DataCompressionType 未指定時，資料壓縮會在建立檔案系統時關閉。

如需詳細資訊，請參閱 [Lustre 資料壓縮](#)。

更新政策：此設定可以在更新期間變更。

ExportPath (選用, String)

Amazon S3 中匯出 FSx for Lustre 檔案系統根目錄的路徑。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的相同 Amazon S3 儲存貯體 ImportPath。您可以提供選用的字首，從 FSx for Lustre 檔案系統匯出新的和變更的資料。如果未提供 ExportPath 值，FSx for Lustre 會設定預設匯出路徑 `s3://amzn-s3-demo-bucket/FSxLustre[creation-timestamp]`。時間戳記使用 UTC 格式，例如 `s3://amzn-s3-demo-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。如果您只指定儲存貯體名稱，例如 `s3://amzn-s3-demo-bucket`，您會取得檔案系統物件到 Amazon S3 儲存貯體物件的 1:1 映射。此映射表示匯出時會覆寫 Amazon S3 中的輸入資料。如果您在匯出路徑中提供自訂字首，例如 `s3://amzn-s3-demo-bucket/[custom-optional-prefix]`，FSx for Lustre 會將檔案系統的內容匯出到 Amazon S3 儲存貯體中的匯出字首。

Note

使用 PERSISTENT_2 部署類型的檔案系統不支援此參數。如《Amazon FSx for Lustre 使用者指南》中的 [將檔案系統連結至 S3 儲存貯體](#) 中所述，設定資料儲存庫關聯。

更新政策：如果變更此設定，則不允許更新。

ImportPath (選用, String)

您用作 FSx for Lustre 檔案系統資料儲存庫的 Amazon S3 儲存貯體路徑（包括選用字首）。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://amzn-s3-demo-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。如《Amazon FSx for Lustre 使用者指南》中的 [將檔案系統連結至 S3 儲存貯體](#) 中所述，設定資料儲存庫關聯。

更新政策：如果變更此設定，則不允許更新。

WeeklyMaintenanceStartTime (選用, String)

執行每週維護的偏好開始時間。其 "d:HH:MM" 格式為 UTC+0 時區。對於此格式，d 是從 1 到 7 的工作日編號，從星期一開始，以星期日結束。此欄位需要引號。

[更新政策：此設定可以在更新期間變更。](#)

AutomaticBackupRetentionDays (選用, Integer)

自動備份的保留天數。將此設定為 0 會停用自動備份。支援的範圍為 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](#)。

[更新政策：此設定可以在更新期間變更。](#)

CopyTagsToBackups (選用, Boolean)

如果為 true，請將 FSx for Lustre 檔案系統的標籤複製到備份。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起始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。

[更新政策：如果變更此設定，則不允許更新。](#)

DailyAutomaticBackupStartTime (選用, String)

週期性每日時間，HH:MM 格式為。HH 是一天中的零填充小時 (00-23)。MM 是一小時的零填充分鐘 (00-59)。例如，05:00 每天指定 5 A.M.。此設定僅適用於 PERSISTENT_1 和 PERSISTENT_2 部署類型。

[更新政策：此設定可以在更新期間變更。](#)

PerUnitStorageThroughput (PERSISTENT_1 和 PERSISTENT_2 部署類型為必要項目, Integer)

以 MB/s/TiB 為單位，說明每 1 TB 儲存的讀取和寫入輸送量。檔案系統輸送量的計算方式是將網路系統儲存容量 (TiB) 乘以 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。這對應到 [PerUnitStorageThroughput](#) 屬性。

有效值：

PERSISTENT_1 SSD 儲存體：50、100、200 MB/s/TiB。

PERSISTENT_1 HDD 儲存：12、40 MB/s/TiB。

PERSISTENT_2 SSD 儲存：125、250、500、1000 MB/s/TiB。

更新政策：如果變更此設定，則不允許更新。

BackupId (選用, String)

指定要用來從現有備份還原 FSx for Lustre 檔案系統的備份 ID。指定 BackupId 設定時，不得指定 AutoImportPolicy、DeploymentType、ExportPath、KmsKeyIdImportPath、StorageCapacity和 ImportedFileChunkSizePerUnitStorageThroughput 設定。這些設定是從備份讀取。此外，不得指定 AutoImportPolicy、ImportPath、ExportPath 和 ImportedFileChunkSize 設定。這對應至 [BackupId](#) 屬性。

更新政策：如果變更此設定，則不允許更新。

KmsKeyId (選用, String)

用於加密 FSx for Lustre 檔案系統資料以用於持久性 FSx for Lustre 檔案系統靜態的 AWS Key Management Service (AWS KMS) 金鑰 ID。如果未指定，則會使用 FSx for Lustre 受管金鑰。SCRATCH_1 和 SCRATCH_2 FSx for Lustre 檔案系統一律使用 FSx for Lustre 受管金鑰進行靜態加密。如需詳細資訊，請參閱 AWS Key Management Service API 參考中的 [加密](#)。

更新政策：如果變更此設定，則不允許更新。

FileSystemId (選用, String)

指定現有 FSx for Lustre 檔案系統的 ID。

如果指定此選項，FsxLustreSettings 則只會使用 中的 MountDir 和 FileSystemId 設定。FsxLustreSettings 會忽略 中的所有其他設定。

Note

如果使用 AWS Batch 排程器，FSx for Lustre 僅適用於前端節點。

Note

檔案系統必須與允許透過連接埠 988、1021、1022 和 1023 傳入和傳出 TCP 流量的安全群組相關聯。

執行下列其中一項動作，確保叢集和檔案系統之間允許流量：

- 設定檔案系統的安全群組，以允許進出叢集子網路 CIDR 或字首清單的流量。

Note

AWS ParallelCluster 會驗證連接埠是否開啟，以及 CIDR 或字首清單是否已設定。AWS ParallelCluster 不會驗證 CIDR 區塊或字首清單的內容。

- 使用 [SlurmQueues](#) // [Networking SecurityGroups](#) 和 [HeadNode](#) // [Networking](#) 設定叢集節點的自訂安全群組 [SecurityGroups](#)。自訂安全群組必須設定為允許叢集與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 只會驗證連接埠是否開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新政策：如果變更此設定，則不允許更新。

AutoImportPolicy (選用, 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。使用此屬性來選擇當您在連結的 Amazon S3 儲存貯體中新增或修改物件時，FSx for Lustre 如何讓您的檔案和目錄清單保持最新狀態。AutoImportPolicy 可以具有下列值：

- NEW - 自動匯入已開啟。FSx for Lustre 會自動匯入任何新增至連結 Amazon S3 儲存貯體且目前不存在於 FSx for Lustre 檔案系統中的新物件的目錄清單。
- NEW_CHANGED - 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，以及在您選擇此選項後在 Amazon S3 儲存貯體中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

3.1.1 版中 NEW_CHANGED_DELETED AWS ParallelCluster 已新增的支援。

如果 AutoImportPolicy 未指定，則自動匯入會關閉。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[從 S3 儲存貯體自動匯入更新](#)。

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](#)。

更新政策：如果變更此設定，則不允許更新。

DriveCacheType (選用, String)

指定檔案系統具有 SSD 磁碟機快取。只有在 StorageType 設定設為 HDD 且 DeploymentType 設定設為 時，才能設定此選項 PERSISTENT_1。這對應到 [DriveCacheType](#) 屬性。如需詳細資訊，請參閱《Amazon [FSx for Lustre 使用者指南](#)》中的 [FSx for Lustre 部署選項](#)。FSx

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。

更新政策：如果變更此設定，則不允許更新。

StorageType (選用, String)

設定您要建立之 FSx for Lustre 檔案系統的儲存類型。有效值為 SSD 和 HDD。

- 設為 SSD 以使用固態硬碟儲存體。
- 設為 HDD 以使用硬碟儲存。PERSISTENT 部署類型支援 HDD。

預設值為 SSD。如需詳細資訊，請參閱《Amazon FSx for Windows 使用者指南》中的[儲存類型選項](#)和《Amazon FSx for Lustre 使用者指南》中的[多個儲存選項](#)。

更新政策：如果變更此設定，則不允許更新。

DeletionPolicy (選用, String)

指定從叢集移除檔案系統或刪除叢集時，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 [DeletionPolicy](#) 設定為 時 Delete，如果刪除叢集，或使用叢集更新移除檔案系統，則會刪除包含其資料的受管檔案系統。

如需詳細資訊，請參閱[共用儲存](#)。

更新政策：此設定可以在更新期間變更。

Note

DeletionPolicy 從 3.3.0 AWS ParallelCluster 版開始支援。

DataRepositoryAssociations (選用, String)

DRAs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。

您無法同時使用 FsxLustreSettings 中的 [ExportPath](#) 和 [ImportPath](#) 與使用 DRAs

[更新政策: 此設定可以在更新期間變更。](#)

Name (必要, String)

DRA 的名稱。您在更新設定時使用此名稱。

[更新政策: 如果變更此設定, 則不允許更新。](#)

BatchImportMetaDataOnCreate (選用, 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

[更新政策: 如果變更此設定, 則不允許更新。](#)

DataRepositoryPath (必要, 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://amzn-s3-demo-bucket/myPrefix/`。此路徑指定 S3 資料儲存庫檔案匯出或匯入的位置。

無法與其他 DRAs 重疊

模式: `^[^\u0000\u0085\u2028\u2029\r\n]{3,4357}$`

下限: 3

上限: 4357

[更新政策: 如果變更此設定, 則不允許更新。](#)

FileSystemPath (必要, 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。

無法與其他 DRAs 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。您只能將 "/" 指定為與檔案系統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

更新政策：如果變更此設定，則不允許更新。

ImportedFileChunkSize (選用, 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

更新政策：此設定可以在更新期間變更。

AutoExportPolicy (選用, 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

[更新政策：此設定可以在更新期間變更。](#)

AutoImportPolicy (選用, 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

[更新政策：此設定可以在更新期間變更。](#)

FsxOntapSettings

Note

您必須定義 FsxOntapSettings FsxOntap 是否為指定 [StorageType](#)。

(選用) FSx for ONTAP 檔案系統的設定。

FsxOntapSettings:
VolumeId: *string*

FsxOntapSettings 屬性

VolumeId (必要, String)

指定現有 FSx for ONTAP 系統的磁碟區 ID。

Note

- 如果使用 AWS Batch 排程器，FSx for ONTAP 僅適用於前端節點。
- 如果 FSx for ONTAP 部署類型為 Multi-AZ，請確定前端節點子網路的路由表已正確設定。
- 3.2.0 AWS ParallelCluster 版中已新增 FSx for ONTAP 的支援。
- 檔案系統必須與允許透過連接埠 111、635、2049 和 4046 傳入和傳出 TCP 和 UDP 流量的安全群組相關聯。

執行下列其中一個動作，確保叢集和檔案系統之間允許流量：

- 設定檔案系統的安全群組，以允許進出叢集子網路 CIDR 或字首清單的流量。

Note

AWS ParallelCluster 會驗證連接埠是否開啟，以及 CIDR 或字首清單是否已設定。AWS ParallelCluster 不會驗證 CIDR 區塊或字首清單的內容。

- 使用 [SlurmQueues](#) // [Networking SecurityGroups](#) 和 [HeadNode](#) // [Networking](#) 為叢集節點設定自訂安全群組 [SecurityGroups](#)。自訂安全群組必須設定為允許叢集與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 只會驗證連接埠是否開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新政策：如果變更此設定，則不允許更新。

FsxOpenZfsSettings

Note

您必須定義 FsxOpenZfsSettings FsxOpenZfs 是否為指定 [StorageType](#)。

(選用) FSx for OpenZFS 檔案系統的設定。

```
FsxOpenZfsSettings:  
VolumeId: string
```

更新政策：如果變更此設定，則不允許更新。

FsxOpenZfsSettings 屬性

VolumeId (必要, String)

指定現有 FSx for OpenZFS 系統的磁碟區 ID。

Note

- 如果使用 AWS Batch 排程器，FSx for OpenZFS 只能在前端節點上使用。
- 3.2.0 AWS ParallelCluster 版中已新增對 FSx for OpenZFS 的支援。
- 檔案系統必須與允許透過連接埠 111、2049、20001、20002 和 20003 傳入和傳出 TCP 和 UDP 流量的安全群組相關聯。

執行下列其中一項動作，確保叢集和檔案系統之間允許流量：

- 設定檔案系統的安全群組，以允許進出叢集子網路 CIDR 或字首清單的流量。

Note

AWS ParallelCluster 會驗證連接埠是否開啟，以及 CIDR 或字首清單是否已設定。AWS ParallelCluster 不會驗證 CIDR 區塊或字首清單的內容。

- 使用 // [Networking SecurityGroups](#) 和 [SlurmQueues / HeadNode / Networking](#) 設定叢集節點的自訂安全群組 [SecurityGroups](#)。自訂安全群組必須設定為允許叢集與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 只會驗證連接埠是否開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新政策：如果變更此設定，則不允許更新。

FileCacheSettings

Note

您必須定義 FileCacheSettings FileCache 是否為指定 [StorageType](#)。

(選用) 檔案快取的設定。

[FileCacheSettings](#):

[FileCacheId](#): *string*

更新政策：如果變更此設定，則不允許更新。

FileCacheSettings 屬性

FileCacheId (必要, 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。
- 檔案快取的支援已新增至 3.7.0 AWS ParallelCluster 版。
- 檔案系統必須與允許透過連接埠 988 傳入和傳出 TCP 流量的安全群組相關聯。

執行下列其中一項動作，確保叢集和檔案系統之間允許流量：

- 設定檔案快取的安全群組，以允許進出叢集子網路 CIDR 或字首清單的流量。

Note

AWS ParallelCluster 會驗證連接埠是否開啟，以及 CIDR 或字首清單是否已設定。AWS ParallelCluster 不會驗證 CIDR 區塊或字首清單的內容。

- 使用 // [Networking SecurityGroups](#) 和 [SlurmQueues](#) / [HeadNode](#) / [Networking](#) 為叢集節點設定自訂安全群組 [SecurityGroups](#)。自訂安全群組必須設定為允許叢集與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 只會驗證連接埠是否開啟。AWS ParallelCluster 不會驗證來源和目的地是否已正確設定。

更新政策：如果變更此設定，則不允許更新。

Iam 區段

(選用) 指定叢集的 IAM 屬性。

Iam:

Roles:

LambdaFunctionsRole: *string*

PermissionsBoundary: *string*

ResourcePrefix: *string*

更新政策：此設定可以在更新期間變更。

Iam 屬性

PermissionsBoundary (選用, String)

用於建立之所有角色的許可界限的 IAM 政策 ARN AWS ParallelCluster。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}`。

更新政策：此設定可以在更新期間變更。

Roles (選用)

指定叢集使用的 IAM 角色設定。

更新政策：此設定可以在更新期間變更。

LambdaFunctionsRole (選用, String)

要使用的 IAM 角色 ARN AWS Lambda。這會覆寫連接至所有 Lambda 函數的預設角色，以支援 AWS CloudFormation 自訂資源。Lambda 需要設定為允許擔任角色的委託人。這

不會覆寫用於的 Lambda 函數角色 AWS Batch。格式是 `arn:${Partition}:iam::
${Account}:role/${RoleName}`。

更新政策：此設定可以在更新期間變更。

ResourcePrefix (選用)

指定由 建立之 IAM 資源的路徑或名稱字首 AWS ParallelCluster。

資源字首必須遵循 [IAM 指定的命名規則](#)：

- 名稱最多可包含 30 個字元。
- 名稱只能是沒有斜線 (/) 字元的字串。
- 路徑最多可達 512 個字元。
- 路徑必須以斜線 (/) 開頭和結尾/。它可以在開始和結束斜線 (/) 之間包含多個斜線 (/)。
- 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
```

傳回組態錯誤。路徑必須有兩個 /。字首本身不能有 /。

更新政策：如果變更此設定，則不允許更新。

LoginNodes 區段

Note

3.7.0 AWS ParallelCluster 版中LoginNodes新增了的支援。

(選用) 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  
  AllowedIps: string  
Networking:  
  SubnetIds:  
    - string  
  SecurityGroups:  
    - string  
  AdditionalSecurityGroups:  
    - 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
      AdditionalIamPolicies:
        - Policy: string
```

更新政策：必須停止叢集中的登入節點，才能變更此設定以進行更新。

LoginNodes 屬性

Pools 屬性

定義具有相同資源組態的登入節點群組。從 AWS ParallelCluster 3.11.0 開始，最多可以指定 10 個集區。

```
Pools:
  - Name: string
    Count: integer
    InstanceType: string
    GracetimePeriod: integer
    Image:
      CustomAmi: string
    Ssh:
      KeyName: string
      AllowedIps: string
    Networking:
      SubnetIds:
        - string
      SecurityGroups:
        - string
```

```
AdditionalSecurityGroups:  
  - 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  
  AdditionalIamPolicies:  
    - Policy: string
```

更新政策：可以新增登入節點集區，但移除集區需要停止叢集中的所有登入節點。

Name (必要 String)

指定LoginNodes集區的名稱。這用於標記LoginNodes資源。

更新政策：如果變更此設定，則不允許更新。

Note

從 3.11.0 AWS ParallelCluster 版開始，更新政策為：必須停止集區中的登入節點，才能變更此設定以進行更新。

Count (必要 Integer)

指定要保持作用中的登入節點數目。

更新政策：此設定可以在更新期間變更。

InstanceType (必要 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，可以變更此設定。

Note

從 3.11.0 AWS ParallelCluster 版開始，更新政策為：必須停止集區中的登入節點，才能變更此設定以進行更新。

GracetimePeriod (選用 Integer)

指定要停用登入節點的通知與實際停止事件之間的最短時間，以分鐘為單位。的有效值 GracetimePeriod 從 3 到 120 分鐘。預設值為 10 分鐘。

Note

觸發事件涉及多個 AWS 服務之間的互動。有時，由於 AWS 服務的內部延遲，網路延遲和資訊傳播可能需要一些時間，因此寬限期可能需要比預期更長的時間。

更新政策：此設定可以在更新期間變更。

Image (選用)

定義登入節點的映像組態。

Image:**CustomAmi:** *String***CustomAmi (選用 String)**

指定用來佈建登入節點的自訂 AMI。如果未指定，則值預設為 中指定的值 [HeadNode 區段](#)。

更新政策：如果變更此設定，則不允許更新。

Ssh (選用)

定義登入節點的 ssh 組態。

Ssh:**KeyName:** *string***AllowedIps:** *string***Note**

從 3.11.0 AWS ParallelCluster 版開始，更新政策為：必須停止集區中的登入節點，才能變更此設定以進行更新。

KeyName (選用 String)

指定用於登入節點的 ssh 金鑰。如果未指定，則值預設為 中指定的值 [HeadNode 區段](#)。

更新政策：必須停止集區中的登入節點，才能變更此設定以進行更新。


Important

已棄用 – 組態參數 LoginNodes/Pool/s/Ssh/KeyName 已棄用，且將在未來的版本中移除。CLI 現在會在叢集組態中使用時傳回警告訊息。如需詳細資訊，請參閱 <https://github.com/aws/aws-parallelcluster/issues/6811> ://。

AllowedIps (選用 String)

指定 SSH 連線登入集區中節點的 CIDR 格式 IP 範圍或字首清單 ID。預設值是在前端節點組態中定義的 [AllowedIps](#)，或未指定 0.0.0.0/0。 [HeadNode 區段](#)

更新政策：必須停止集區中的登入節點，才能變更此設定以進行更新。

 Note

3.11.0 AWS ParallelCluster 版中新增了對登入節點的 AllowedIps 支援。

Networking (必要)

Networking:

SubnetIds:


- *string*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

 Note

從 3.11.0 AWS ParallelCluster 版開始，更新政策為：必須停止集區中的登入節點，才能變更此設定以進行更新。

SubnetIds (必要 [String])

您佈建登入節點集區的現有子網路 ID。您只能定義一個子網路。

更新政策：如果變更此設定，則不允許更新。

SecurityGroups (選用 [String])

用於登入節點集區的安全群組清單。如果未指定安全群組，會為您 AWS ParallelCluster 建立安全群組。

更新政策：如果變更此設定，則不允許更新。

AdditionalSecurityGroups (選用 [String])

用於登入節點集區的其他安全群組清單。

更新政策：如果變更此設定，則不允許更新。

Dcv (選用)

定義在[登入節點](#)上執行的 NICE DCV 伺服器的組態設定。如需詳細資訊，請參閱[透過 Amazon DCV 連接至前端和登入節點](#)

Dcv:

[Enabled](#): *boolean*

[Port](#): *integer*

[AllowedIps](#): *string*

Important

根據預設，設定的 NICE DCV 連接埠 AWS ParallelCluster 開放給所有 IPv4 地址。只有在您擁有 NICE DCV 工作階段的 URL，並在從 pcluster dcv-connect 傳回 URL 的 30 秒內連線至 NICE DCV 工作階段時，才能連線至 NICE DCV 連接埠。使用 AllowedIps 設定進一步限制存取具有 CIDR 格式 IP 範圍的 NICE DCV 連接埠，並使用連接埠設定來設定非標準連接埠。

更新政策：如果變更此設定，則不允許更新。

Note

3.11.0 AWS ParallelCluster 版中新增了對登入節點上 DCV 的支援。

Enabled (必要 Boolean)

指定是否在集區中的登入節點上啟用 NICE DCV。預設值為 false。

更新政策：如果變更此設定，則不允許更新。

Note

NICE DCV 會自動產生自我簽署憑證，用於保護在登入節點上執行的 NICE DCV 用戶端和 NICE DCV 伺服器之間的流量。若要設定您自己的憑證，請參閱[Amazon DCV HTTPS 憑證](#)。

Port (選用 Integer)

指定 NICE DCV 的連接埠。預設值為 8443。

更新政策：如果變更此設定，則不允許更新。

AllowedIps (選用 String)

指定連線至 NICE DCV 的 CIDR 格式 IP 範圍。只有在 AWS ParallelCluster 建立安全群組時，才會使用此設定。預設值為 0.0.0.0/0，允許從任何網際網路地址存取。

更新政策：如果變更此設定，則不允許更新。

CustomActions (選用)

指定要在登入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
    OnNodeUpdated:
      Sequence:
        - Script: string
        Args:
          - string
      Script: string
      Args:
        - string
```


Note

3.11.0 AWS ParallelCluster 版中新增了對登入節點上自訂動作的支援。

OnNodeStart (選用)

在啟動任何節點部署引導動作之前，指定要在[登入節點](#)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[自訂引導操作](#)。

Sequence (選用)

要執行的指令碼清單。會以與組態檔案中列出的相同順序 AWS ParallelCluster 執行指令碼，從第一個開始。

Script (必要 String)

指定要使用的檔案。檔案路徑的開頭可以是 https://或 s3://。

Args (選用 [String])

要傳遞至指令碼的引數清單。

更新政策：如果變更此設定，則不允許更新。

Script (必要 String)

指定要用於單一指令碼的檔案。檔案路徑的開頭可以是 https://或 s3://。

Args (選用 [String])

要傳遞至單一指令碼的引數清單。

OnNodeConfigured (選用)

指定節點引導程序完成後，要在[登入節點](#)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[自訂引導操作](#)。

Sequence (選用)

要執行的指令碼清單。會以與組態檔案中列出的相同順序 AWS ParallelCluster 執行指令碼，從第一個開始。

Script (必要 String)

指定要使用的檔案。檔案路徑的開頭可以是 https://或 s3://。

Args (選用 [String])

要傳遞至指令碼的引數清單。

更新政策：如果變更此設定，則不允許更新。

Script (必要 String)

指定要用於單一指令碼的檔案。檔案路徑的開頭可以是 https://或 s3://。

Args (選用 [String])

要傳遞至單一指令碼的引數清單。

更新政策：如果變更此設定，則不允許更新。

OnNodeUpdated (選用)

指定在前端節點更新完成後執行的單一指令碼或一系列指令碼，並將排程器和共用儲存與最新的叢集組態變更保持一致。如需詳細資訊，請參閱[自訂引導操作](#)。

Sequence (選用)

要執行的指令碼清單。會以與組態檔案中列出的相同順序 AWS ParallelCluster 執行指令碼，從第一個開始。

Script (必要 String)

指定要使用的檔案。檔案路徑的開頭可以是 https://或 s3://。

Args (選用 [String])

要傳遞至指令碼的引數清單。

Script (必要 String)

指定要用於單一指令碼的檔案。檔案路徑的開頭可以是 https://或 s3://。

Args (選用 [String])

要傳遞至單一指令碼的引數清單。

更新政策：如果變更此設定，則不允許更新。

Note

AWS ParallelCluster 不支援同時包含單一指令碼和Sequence相同的自訂動作。

Iam (選用)

指定要在登入節點上使用的執行個體角色或執行個體描述檔，以覆寫叢集的預設執行個體角色或執行個體描述檔。

```
Iam:  
  InstanceRole: string  
  InstanceProfile: string  
  AdditionalIamPolicies:  
    - Policy: string
```

Note

從 3.11.0 AWS ParallelCluster 版開始，更新政策為：必須停止集區中的登入節點，才能變更此設定以進行更新。

InstanceProfile (選用 String)

指定執行個體描述檔以覆寫預設的登入節點執行個體描述檔。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`。如果指定此選項，則無法指定 InstanceRole 和 AdditionalIamPolicies 設定。

[更新政策：如果變更此設定，則不允許更新。](#)

InstanceRole (選用 String)

指定執行個體角色以覆寫預設的登入節點執行個體角色。您不能同時指定 InstanceProfile 和 InstanceRole。格式是 `arn:Partition:iam::Account:role/RoleName`。如果指定此選項，則無法指定 InstanceProfile 和 AdditionalIamPolicies 設定。

[更新政策：如果變更此設定，則不允許更新。](#)

AdditionalIamPolicies (選用)

```
AdditionalIamPolicies:  
  - Policy: string
```

IAM 政策 Amazon Resource Name (ARN)。

指定 Amazon EC2 IAM 政策的 Amazon Resource Name (ARNs) 清單。除了 所需的許可之外，此清單還會連接到用於登入節點的根角色 AWS ParallelCluster。

IAM 政策名稱及其 ARN 不同。無法使用名稱。

如果指定此選項，則無法指定 InstanceProfile 和 InstanceRole 設定。我們建議您使用 `AdditionalIamPolicies` 因為 `AdditionalIamPolicies` 已新增至 AWS ParallelCluster 所需的許可，而且 InstanceRole 必須包含所有必要的許可。隨著新增功能，所需的許可通常會從發行版本變更為發行版本。

沒有預設值。

更新政策：如果變更此設定，則不允許更新。

Policy (必要 [String])

更新政策：如果變更此設定，則不允許更新。

Monitoring 區段

(選用) 指定叢集的監控設定。

```
Monitoring:
  Logs:
    CloudWatch:
      Enabled: boolean
      RetentionInDays: integer
      DeletionPolicy: string
    Rotation:
      Enabled: boolean
  Dashboards:
    CloudWatch:
      Enabled: boolean
    DetailedMonitoring: boolean
  Alarms:
    Enabled: boolean
```

更新政策：更新期間不會分析此設定。

Monitoring 屬性

Logs (選用)

叢集的日誌設定。

更新政策：如果變更此設定，則不允許更新。

CloudWatch (選用)

叢集的 CloudWatch Logs 設定。

更新政策：如果變更此設定，則不允許更新。

Enabled (必要, Boolean)

如果為 true，叢集日誌會串流到 CloudWatch Logs。預設值為 true。

更新政策：如果變更此設定，則不允許更新。

RetentionInDays (選用, Integer)

在 CloudWatch Logs 中保留日誌事件的天數。預設值為 180。支援的值為 0、1、3、5、7、14、30、60、90、120、150、180、365、400、545、731、1827 和 3653。值 0 將使用預設 CloudWatch 日誌保留設定，即永不過期。

更新政策：此設定可以在更新期間變更。

DeletionPolicy (選用, String)

指出是否在刪除叢集時刪除 CloudWatch Logs 上的日誌事件。可能的值為 Delete 和 Retain。預設值為 Retain。

更新政策：此設定可以在更新期間變更。

Rotation (選用)

叢集的日誌輪換設定。

更新政策：如果變更此設定，則不允許更新。

Enabled (必要, Boolean)

如果為 true，則會啟用日誌輪換。預設值為 true。當 AWS ParallelCluster 設定的日誌檔案達到特定大小時，它會輪換並維護單一備份。如需詳細資訊，請參閱 [AWS ParallelCluster 設定的日誌輪換](#)。

更新政策：如果變更此設定，則不允許更新。

Dashboards (選用)

叢集的儀表板設定。

更新政策：此設定可以在更新期間變更。

CloudWatch (選用)

叢集的 CloudWatch 儀表板設定。

更新政策：此設定可以在更新期間變更。

Enabled (必要, Boolean)

如果為 true，則會啟用 CloudWatch 儀表板。預設值為 true。

更新政策：此設定可以在更新期間變更。

DetailedMonitoring (選用, Boolean)

如果設定為 true，則會為運算機群 Amazon EC2 執行個體啟用詳細監控。啟用時，Amazon EC2 主控台會顯示圖形，以每隔 1 分鐘監控執行個體。啟用此功能時會產生額外的成本。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](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

Note

DetailedMonitoring 從 3.6.0 AWS ParallelCluster 版開始新增。

Alarms (選用)


叢集的 CloudWatch 警示。

更新政策：此設定可以在更新期間變更。

Enabled (選用)

如果為 true，則會建立叢集的 CloudWatch 警示。預設值為 true。

更新政策：此設定可以在更新期間變更。

 Note

從 3.8.0 AWS ParallelCluster 版開始，會針對 Head Node 建立下列警示：Amazon EC2 運作狀態檢查、CPU/Memory/Disk使用率，以及包含所有其他警示的複合警示。

Tags 區段

(選用)，Array 定義由使用 CloudFormation 並傳播至所有叢集資源的標籤。如需詳細資訊，請參閱AWS CloudFormation 《使用者指南》中的[CloudFormation 資源標籤](#)。

Tags:

- Key: *string*
- Value: *string*

更新政策：此設定可以在更新期間變更。

Tags 屬性

Key (必要, String)


定義標籤的名稱。

更新政策：此設定可以在更新期間變更。

Value (必要, String)

定義標籤的值。

更新政策：此設定可以在更新期間變更。

 Note

從 AWS ParallelCluster 3.15.0 開始，支援標籤更新，但有下列限制：

- HeadNode 上的 EBS 磁碟區 - 只會保留叢集建立時來自的標籤；不支援更新此 EBS 磁碟區的標籤。
- 執行中的節點 - 標籤更新不會套用至執行中的運算或登入節點。

AdditionalPackages 區段

(選用) 用來識別要安裝的其他套件。

AdditionalPackages:

IntelSoftware:

IntelHpcPlatform: *boolean*

更新政策：如果變更此設定，則不允許更新。

IntelSoftware

(選用) 定義 Intel Select 解決方案的組態。

IntelSoftware:

IntelHpcPlatform: *boolean*

更新政策：如果變更此設定，則不允許更新。

IntelSoftware 屬性

IntelHpcPlatform (選用, Boolean)

如果為 true，表示接受 Intel Parallel Studio [的最終使用者授權合約](#)。這會導致 Intel Parallel Studio 安裝在前端節點上並與運算節點共用。這會為前端節點引導所需的時間新增幾分鐘。

更新政策：如果變更此設定，則不允許更新。

Note

從 3.10.0 AWS ParallelCluster 版開始，不再支援 IntelHpcPlatform 參數。

DirectoryService 區段

Note

3.1.1 AWS ParallelCluster 版中DirectoryService已新增 的支援。

(選用) 支援多個使用者存取之叢集的目錄服務設定。

AWS ParallelCluster 會透過 [System Security Services 協助程式 \(SSSD\)](#) 支援的輕量型目錄存取通訊協定 (LDAP)，管理支援多個使用者存取叢集的許可。如需詳細資訊，請參閱《AWS Directory Service 管理指南》中的[什麼是 AWS Directory Service ?](#)。

我們建議您透過 TLS/SSL 使用 LDAP (簡短的縮寫 LDAPS)，以確保透過加密管道傳輸任何潛在的敏感資訊。

```
DirectoryService:
  DomainName: string
  DomainAddr: string
  PasswordSecretArn: string
  DomainReadOnlyUser: string
  LdapTlsCaCert: string
  LdapTlsReqCert: string
  LdapAccessFilter: string
  GenerateSshKeysForUsers: boolean
  AdditionalSssdConfigs: dict
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

DirectoryService 屬性

Note

如果您打算在沒有網際網路存取的單一子網路 AWS ParallelCluster 中使用，請參閱 [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#) 以取得其他需求。

DomainName (必要, String)

您用於身分資訊的 Active Directory (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 DN 範例：DC=*corp*,DC=*example*,DC=*com*

此屬性對應於稱為的 sssd-ldap 參數ldap_search_base。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

DomainAddr (必要, String)

指向做為 LDAP 伺服器的 AD 網域控制器的 URI 或 URIs。URI 對應至稱為的 SSSD-LDAP 參數 `ldap_uri`。此值可以是以逗號分隔的 URIs 字串。若要使用 LDAP，您必須將 `ldap://` 新增至每個 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
```

如果您使用 LDAPS 進行憑證驗證，URIs 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，URIs 可以是主機名稱或 IP 地址。

使用 LDAP over TLS/SSL (LDAPS) 來避免透過未加密的頻道傳輸密碼和其他敏感資訊。如果 AWS ParallelCluster 找不到通訊協定，它會 `ldaps://` 新增到每個 URI 或主機名稱的開頭。

[更新政策：必須停止運算機群，才能變更此設定以進行更新。](#)

PasswordSecretArn (必要, String)

包含 `DomainReadOnlyUser` 純文字密碼之 AWS Secrets Manager 秘密的 Amazon Resource Name (ARN)。秘密的內容對應至稱為的 SSSD-LDAP 參數 `ldap_default_authtok`。

Note

當您使用 AWS Secrets Manager 主控台建立秘密時，請務必選取「其他秘密類型」、選取純文字，並只在秘密中包含密碼文字。

如需如何使用 AWS Secrets Manager 建立秘密的詳細資訊，請參閱 [建立 AWS Secrets Manager 秘密](#)

當 LDAP 用戶端請求身分資訊 `DomainReadOnlyUser` 時，會使用密碼向 AD 網域進行身分驗證。

如果使用者具有的許可 [DescribeSecret](#)，`PasswordSecretArn` 則會驗證。如果指定的秘密存在，`PasswordSecretArn` 則有效。如果使用者 IAM 政策不包含

DescribeSecret , PasswordSecretArn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使用者政策](#)。

當秘密的值變更時 , 叢集不會自動更新。若要更新叢集的新秘密值 , 您必須使用 [the section called "pcluster update-compute-fleet"](#) 命令停止運算機群 , 然後從前端節點內執行下列命令。

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

[更新政策 : 必須停止運算機群 , 才能變更此設定以進行更新。](#)

DomainReadOnlyUser (必要 , String)

驗證叢集使用者登入時 , 用於查詢 AD 網域身分資訊的身分。它對應於稱為的 SSSD-LDAP 參數ldap_default_bind_dn。將您的 AD 身分資訊用於此值。

以節點上特定 LDAP 用戶端所需的格式指定身分 :

- MicrosoftAD :

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

- SimpleAD :

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

[更新政策 : 必須停止運算機群 , 才能變更此設定以進行更新。](#)

LdapTlsCaCert (選用 , String)

憑證套件的絕對路徑 , 其中包含憑證鏈中為網域控制站發出憑證的每個憑證授權單位的憑證。它對應於稱為的 SSSD-LDAP 參數ldap_tls_cacert。

憑證套件是由以 PEM 格式串連不同憑證的檔案 , 也稱為 Windows 中的 DER Base64 格式。它用於驗證充當 LDAP 伺服器的 AD 網域控制器身分。

AWS ParallelCluster 不負責在節點上初始放置憑證。身為叢集管理員 , 您可以在建立叢集後手動設定前端節點中的憑證 , 也可以使用[引導指令碼](#)。或者 , 您可以使用 Amazon Machine Image (AMI) , 其中包含在前端節點上設定的憑證。

[Simple AD](#) 不提供 LDAPS 支援。若要了解如何將 Simple AD 目錄與整合 AWS ParallelCluster , 請參閱 AWS 安全部落格中的[如何設定 Simple AD 的 LDAPS 端點](#)。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

LdapTlsReqCert (選用, String)

指定要在 TLS 工作階段中對伺服器憑證執行哪些檢查。它對應於稱為的 SSSD-LDAP 參數 `ldap_tls_reqcert`。

有效值：`never`、`allow`、`try`、`demand` 和 `hard`。

`never` 即使發現憑證有問題，`allow`、`try` 和 `demand` 也會 `try` 啟用連線以繼續。

`demand` 如果找不到憑證問題，`try` 和 `hard` 會啟用通訊以繼續。

如果叢集管理員使用不需要憑證驗證才能成功的值，則會將警告訊息傳回給管理員。基於安全考量，建議您不要停用憑證驗證。

預設值為 `hard`。

更新政策：必須停止運算機群，才能變更此設定以進行更新。

LdapAccessFilter (選用, String)

指定篩選條件，以限制對使用者子集的目錄存取。此屬性對應至稱為的 SSSD-LDAP 參數 `ldap_access_filter`。您可以使用它來將查詢限制為支援大量使用者的 AD。

此篩選條件可封鎖使用者存取叢集。不過，這不會影響封鎖使用者的可探索性。

如果設定此屬性，則 SSSD 參數 `access_provider` 會由 AWS ParallelCluster `ldap` 內部設定為 `ad`，且不得由 [DirectoryService AdditionalSssdConfigs](#) 設定修改。

如果省略此屬性，且未在 [DirectoryService](#) / `AdditionalSssdConfigs` 中指定自訂使用者存取權 `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 an alias that starts with
"SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias that starts with
"SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

GenerateSshKeysForUsers (選用, Boolean)

定義 是否在前端節點的初始身分驗證之後，立即為叢集使用者 AWS ParallelCluster 產生 SSH 金鑰。

如果設為 `true`，如果 SSH 金鑰不存在 `USER_HOME_DIRECTORY/.ssh/id_rsa`，則在前端節點上進行第一次身分驗證後，每個使用者都會產生 SSH 金鑰並儲存到。

對於尚未在前端節點上進行身分驗證的使用者，在下列情況下可能會發生第一次身分驗證：

- 使用者第一次使用自己的密碼登入前端節點。
- 在前端節點中，`sudoer` 第一次切換到使用者：`su USERNAME`
- 在前端節點中，`sudoer` 第一次以使用者身分執行命令：`su -u USERNAME COMMAND`

使用者可以使用 SSH 金鑰來後續登入叢集前端節點和運算節點。使用時 AWS ParallelCluster，設計會停用叢集運算節點的密碼登入。如果使用者尚未登入前端節點，則不會產生 SSH 金鑰，而且使用者將無法登入運算節點。

預設值為 `true`。

[更新政策：必須停止運算機群，才能變更此設定以進行更新。](#)

AdditionalSssdConfigs (選用, Dict)

索引鍵/值對的字典，其中包含要寫入叢集執行個體上 SSSD 組態檔案的 SSSD 參數和值。如需 SSSD 組態檔案的完整說明，請參閱 SSSD 和相關組態檔案的執行個體手冊頁面。

SSSD 參數和值必須與 SSSD AWS ParallelCluster 組態相容，如下列清單所述。

- `id_provider` 由 `ldap` 內部設定為 AWS ParallelCluster，且不得修改。
- `access_provider` 當指定 `/ AWS ParallelCluster` 時，[LdapAccessFilter](#) 會在 `ldap` 內部設定為 [DirectoryService](#)，且不得修改此設定。

如果 [LdapAccessFilter](#) 省略 [DirectoryService](#) /，也會省略其 `access_provider` 規格。例如，如果您在 `simple` 中 `access_provider` 將設定為 [AdditionalSssdConfigs](#)，則 [LdapAccessFilter](#) 不得指定 [DirectoryService](#) /。

下列組態程式碼片段是有效組態的範例 `AdditionalSssdConfigs`。

此範例會啟用 SSSD 日誌的偵錯層級、將搜尋基礎限制為特定組織單位，以及停用登入資料快取。

```
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](#) 的組態 `access_provider`。來自的使用者 `EngineeringTeam` 可以存取目錄。在此情況下 `LdapAccessFilter`，不得設定 `DirectoryService` /。

```
DirectoryService:
  ...
  AdditionalSssdConfigs:
    access_provider: simple
    simple_allow_groups: EngineeringTeam
```

更新政策：必須停止運算機群，才能變更此設定以進行更新。

DeploymentSettings 區段

Note

DeploymentSettings 從 3.4.0 AWS ParallelCluster 版開始新增。

(選用) 指定部署設定組態。

```
DeploymentSettings:
  LambdaFunctionsVpcConfig:
    SecurityGroupIds:
      - string
    SubnetIds:
      - string
  DisableSudoAccessForDefaultUser: Boolean
  DefaultUserHome: string # 'Shared' or 'Local'
```

DeploymentSettings 屬性

LambdaFunctionsVpcConfig

(選用) 指定 AWS Lambda 函數 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 (必要, [String])


連接至 Lambda 函數的 Amazon VPC 安全群組 IDs 清單。

更新政策：如果變更此設定，則不允許更新。

SubnetIds (必要, [String])


連接至 Lambda 函數的子網路 IDs 清單。

更新政策：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 屬性

 Note

只有 Slurm 叢集才支援此組態選項。

(選用) 如果 True 為，預設使用者的 sudo 權限將會停用。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，您必須停止運算機群和所有登入節點。

更新政策：必須停止運算機群和登入節點，才能變更此設定以進行更新。

DefaultUserHome 屬性

設為 `Shared`，叢集將使用預設設定，並透過在整個叢集中共用預設使用者的目錄 `/home/<default user>`。

當設定為 `Local`，前端節點、登入節點和運算節點將各自具有儲存在 `local` 中的個別本機預設使用者目錄 `local/home/<default user>`。

建置映像組態檔案

AWS ParallelCluster 第 3 版使用 YAML 1.1 檔案來建置映像組態參數。請確認縮排正確，以減少組態錯誤。如需詳細資訊，請參閱位於的 YAML [1](https://yaml.org/spec/1.1/) <https://yaml.org/spec/1.1/> 規格。

這些組態檔案用於定義如何使用 EC2 Image Builder 建置您的 AWS ParallelCluster AMIs。使用 `pcluster build-image` 命令觸發自訂 AMI 建置程序。如需範例組態檔案，請參閱 https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema。

主題

- [建置映像組態檔案屬性](#)
- [Build 區段](#)
- [Image 區段](#)
- [DeploymentSettings 區段](#)

建置映像組態檔案屬性

Region (選用, String)

指定 `build-image` 操作 AWS 區域的。例如 `us-east-2`。

CustomS3Bucket (選用, String)

指定在您的 AWS 帳戶中建立的 Amazon S3 儲存貯體名稱，以存放自訂 AMI 建置程序所使用的資源，以及匯出日誌。映像使用的資訊位於映像 `config` 的自訂儲存貯體中。會在您建立叢集的每個 AWS 區域中 AWS ParallelCluster 維護一個 Amazon S3 儲存貯體。根據預設，這些 Amazon S3 儲存貯體會命名為 `parallelcluster-hash-v1-DO-NOT-DELETE`。

Build 區段

(必要) 指定要在其中建置映像的組態。

```
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
  Installation:
    NvidiaSoftware:
      Enabled: boolean
    LustreClient:
      Enabled: boolean
```

Build 屬性

InstanceType (必要, String)

指定用於建置映像之執行個體的執行個體類型。

SubnetId (選用, String)

指定要佈建執行個體以建置映像的現有子網路 ID。提供的子網路需要網際網路存取。請注意，如果建置失敗，[您可能需要修改子網路的 IP 定址屬性](#)。

⚠ Warning

`pcluster build-image` 使用預設 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。

當您指定 `SubnetId` 時，建議您也指定 `SecurityGroupIds` 屬性。如果您離開 `SecurityGroupIds`，AWS ParallelCluster 將使用預設安全群組或依賴指定子網路中的預設行為。當您同時使用兩者時，您會獲得這些優勢：

- 精細控制：當您明確定義兩者時，可確保在映像建置程序期間啟動的執行個體放置在正確的子網路中，並擁有建置元件和任何必要服務的精確網路存取（例如針對建置指令碼存取 S3）。
- 安全最佳實務：當您定義適當的安全群組時，這有助於限制網路只能存取必要的連接埠和服務，進而增強建置環境的安全性。
- 避免潛在問題：如果您只依賴預設值，這可能會導致安全群組太開放或太嚴格，這可能會在建置過程中導致問題。

`ParentImage` (必要, String)

指定基礎映像。父系映像可以是非 AWS ParallelCluster AMI 或相同版本的官方 AWS ParallelCluster AMI。您無法從不同版本的 使用 AWS ParallelCluster 官方或自訂 AMI AWS ParallelCluster。格式必須是映像的 ARN `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion` 或 AMI ID `ami-12345678`。

`SecurityGroupIds` (選用, [String])

指定映像的安全群組 IDs 清單。

Ims

Ims 屬性

(選用) 指定 Amazon EC2 ImageBuilder 建置和測試執行個體中繼資料服務 (IMDS) 設定。

```
Ims:  
ImsSupport: string
```

ImdsSupport (選用, String)

指定在 Amazon EC2 ImageBuilder 建置和測試執行個體中支援哪些 IMDS 版本。支援的值為 v2.0 和 v1.0。預設值為 v2.0。

如果 ImdsSupport 設定為 v1.0，則同時支援 IMDSv1 和 IMDSv2。

如果 ImdsSupport 設定為 v2.0，則僅支援 IMDSv2。

如需詳細資訊，請參閱《Amazon EC2 [IMDSv2](#)》。 Amazon EC2

更新政策：如果變更此設定，則不允許更新。

Note

從 3.7.0 AWS ParallelCluster 版開始，ImdsSupport 預設值為 v2.0。我們建議您在自訂動作呼叫中 ImdsSupport 將設定為 v2.0，並將 IMDSv1 取代為 IMDSv2。
3.3.0 AWS ParallelCluster 版 [ImdsSupport](#) 新增了對 [Imds](#) / 的支援。

Iam

Iam 屬性

(選用) 指定映像建置的 IAM 資源。

Iam:

```
InstanceRole: string  
InstanceProfile: string  
CleanupLambdaRole: string  
AdditionalIamPolicies:  
  - Policy: string  
PermissionsBoundary: string
```

InstanceProfile (選用, String)

指定執行個體描述檔以覆寫 EC2 Image Builder 執行個體的預設執行個體描述檔。

InstanceProfile InstanceRole 和 AdditionalIamPolicies 無法一起指定。格式是 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`。

InstanceRole (選用, 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 InstanceRole 和 AdditionalIamPolicies 無法一起指定。格式是 `arn:Partition:iam::Account:role/RoleName`。

CleanupLambdaRole (選用, String)

用於 AWS Lambda 函數的 IAM 角色 ARN，該函數支援 CloudFormation 自訂資源，可在建置完成時移除建置成品。Lambda 需要設定為允許擔任角色的委託人。格式是 `arn:Partition:iam::Account:role/RoleName`。

AdditionalIamPolicies (選用)

指定要連接至用於產生自訂 AMI 之 EC2 Image Builder 執行個體的其他 IAM 政策。

```
AdditionalIamPolicies:  
- Policy: string
```

Policy (選用, [String])

IAM 政策清單。格式是 `arn:Partition:iam::Account:policy/PolicyName`。

PermissionsBoundary (選用, String)

要用作由 建立之所有角色之許可界限的 IAM 政策 ARN AWS ParallelCluster。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](#)。格式是 `arn:Partition:iam::Account:policy/PolicyName`。

Components

Components 屬性

(選用) 除了預設提供的元件之外，指定要在 AMI 建置程序期間使用的 Amazon EC2 ImageBuilder 元件 AWS ParallelCluster。這類元件可用來自訂 AMI 建置程序。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。

```
Components:  
- Type: string  
  Value: string
```

Type (選用, String)

指定元件的類型值對類型。類型可以是 `arn` 或 `script`。

Value (選用, String)

指定元件的類型值對的值。當類型為 `arn`，這是 EC2 Image Builder 元件的 ARN。當類型為 `script`，這是指向指令碼的 `https` 或 `s3` 連結，以便在您建立 EC2 Image Builder 元件時使用。

Tags

Tags 屬性

(選用) 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 (選用, String)

定義標籤的名稱。

Value (選用, String)

定義標籤的值。

UpdateOsPackages

UpdateOsPackages 屬性

(選用) 指定是否在安裝 AWS ParallelCluster 軟體堆疊之前更新作業系統。

UpdateOsPackages:

- Enabled: *boolean*

Enabled (選用, Boolean)

如果為 `true`，作業系統會在安裝 AWS ParallelCluster 軟體之前更新並重新啟動。預設值為 `false`。

Note

啟用 UpdateOsPackages 時，所有可用的作業系統套件都會更新，包括核心。身為客戶，您有責任確認更新與更新中未包含的 AMI 相依性相容。

例如，假設您想要為核心 AWS ParallelCluster 版本 Y.0 和某些元件版本 Z.0 隨附的版本 X.0 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 支援核心 Y.1。

Installation

Installation 屬性

(選用) 指定要在映像上安裝的其他軟體。

```
Installation:  
  NvidiaSoftware:  
    Enabled: boolean  
  LustreClient:  
    Enabled: boolean
```

NvidiaSoftware 屬性 (選用)

指定要安裝的 Nvidia 軟體。

```
NvidiaSoftware:  
  Enabled: boolean
```

Enabled (選用, boolean)

如果為 true，則會安裝 Nvidia GPU 驅動程式和 CUDA。預設值為 false。

LustreClient 屬性 (選用)

指定將安裝 Amazon FSx Lustre 用戶端。

```
LustreClient:  
  Enabled: boolean
```

Enabled (選用, boolean)

如果為 true，則會安裝 Lustre 用戶端。預設值為 true。

Image 區段

(選用) 定義映像建置的映像屬性。

```
Image:
  Name: string
  RootVolume:
    Size: integer
    Encrypted: boolean
    KmsKeyId: string
  Tags:
    - Key: string
      Value: string
```

Image 屬性

Name (選用, String)

指定 AMI 的名稱。如果未指定，則會使用呼叫 [pcluster build-image](#) 命令時使用的名稱。

Tags

Tags 屬性

(選用) 指定影像的鍵/值對。

```
Tags:
  - Key: string
    Value: string
```

Key (選用, String)

定義標籤的名稱。

Value (選用, String)

定義標籤的值。

RootVolume

RootVolume 屬性

(選用) 指定映像根磁碟區的屬性。

```
RootVolume:  
  Size: integer  
  Encrypted: boolean  
  KmsKeyId: string
```

Size (選用 , Integer)

指定映像根磁碟區的大小，以 GiB 為單位。預設大小為 [ParentImage](#) 加 27 GiB 的大小。

Encrypted (選用 , Boolean)

指定磁碟區是否已加密。預設值為 `false`。

KmsKeyId (選用 , String)

指定用於加密磁碟區的 AWS KMS 金鑰 ARN。格式為
「arn:*Partition*:kms:*Region*:*Account*:key/*KeyId*。

DeploymentSettings 區段

Note

DeploymentSettings 從 3.4.0 AWS ParallelCluster 版開始新增。

(選用) 指定部署設定組態。

```
DeploymentSettings:  
  LambdaFunctionsVpcConfig:  
    SecurityGroupIds:  
      - string  
    SubnetIds:  
      - string
```


DeploymentSettings 屬性

LambdaFunctionsVpcConfig

(選用) 指定 AWS Lambda 函數 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 (必要, [String])

連接至 Lambda 函數的 Amazon VPC 安全群組 IDs 清單。

更新政策：如果變更此設定，則不允許更新。

SubnetIds (必要, [String])

連接至 Lambda 函數的子網路 IDs 清單。

更新政策：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。

AWS ParallelCluster API 參考

本節提供每個 AWS ParallelCluster API 動作的說明、語法和使用範例。

主題

- [buildImage](#)
- [createCluster](#)
- [deleteCluster](#)

- [deleteClusterInstances](#)
- [deleteImage](#)
- [describeCluster](#)
- [describeClusterInstances](#)
- [describeComputeFleet](#)
- [describeImage](#)
- [getClusterLogEvents](#)
- [getClusterStackEvents](#)
- [getImageLogEvents](#)
- [getImageStackEvents](#)
- [listClusters](#)
- [listClusterLogStreams](#)
- [listImageLogStreams](#)
- [listImages](#)
- [listOfficialImages](#)
- [updateCluster](#)
- [updateComputeFleet](#)

buildImage

在 中建立自訂 AWS ParallelCluster 映像 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

imageConfiguration

影像組態做為 YAML 文件。

類型：字串

必要：是

imageId

要建置之映像的 ID。

類型：字串

必要：是

dryrun

如果設定為 true，則只執行請求驗證，而不建立任何資源。使用此參數來驗證映像組態。預設值為 false。

類型：布林值

必要：否

region

您在 AWS 區域 其中執行 命令以建置映像的。

類型：字串

必要：否

rollbackOnFailure

如果設定為 `true`，則映像堆疊復原會在映像無法建立時發生。預設值為 `false`。

類型：布林值

必要：否

suppressValidators

識別要隱藏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

導致映像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 ID。

類型：字串

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (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

ec2AmiInfo

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

建置映像 AWS 區域的。

類型：字串

version

用來建置映像的 AWS ParallelCluster 版本。

類型：字串

validationMessages

驗證層級低於 validationFailureLevel 的訊息清單validationFailureLevel。訊息清單會在組態驗證期間收集。

id

驗證器 ID。

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)
```

200 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

clusterConfiguration

叢集組態做為 YAML 文件。

類型：字串

必要：是

clusterName

要建立的叢集名稱。

名稱必須以字母字元開頭。名稱最多可有 60 個字元。如果啟用Slurm會計，名稱最多可有 40 個字元。

類型：字串

必要：是

dryrun

如果設為 true，則只會執行請求驗證，但不會建立任何資源。使用此參數來驗證叢集組態。預設值為 false。

類型：布林值

必要：否

region

AWS 區域 叢集所在的。

類型：字串

必要：否

rollbackOnFailure

如果設為 `true`，則叢集堆疊復原會在叢集無法建立時發生。預設值為 `true`。

類型：布林值

必要：否

suppressValidators

識別要隱藏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

導致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

clusterName

叢集的名稱。

類型：字串

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (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

clusterStatus

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

AWS 區域 叢集建立所在的。

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

validation_messages

驗證層級低於 `validationFailureLevel` 的訊息清單。訊息清單會在組態驗證期間收集。

id

驗證器的 ID。

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

驗證器的類型。

類型：字串

範例

Python

請求

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

200 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"
}
```

deleteCluster

開始刪除叢集。

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

clusterName

叢集的名稱。

類型：字串

必要：是

region

刪除叢集 AWS 區域的。

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

叢集執行個體的清單。

clusterName

叢集的名稱。

類型：字串

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (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

clusterStatus

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

建立叢集 AWS 區域的。

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)
```

200 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

啟動強制終止所有叢集運算節點。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

clusterName

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

AWS 區域 叢集所在的。

類型：字串

必要：否

回應內文

無

範例

Python

請求

```
$ delete_cluster_instances(cluster_name_3x)
```

200 個回應

無

deleteImage

啟動刪除自訂 AWS ParallelCluster 映像。

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 ID。

類型：字串

必要：是

force

如果設定為 true，請強制刪除 AMI。如果有使用 AMI 的執行個體，或共用 AMI，請使用此參數。
預設值為 false。

類型：布林值

必要：否

region

建立映像 AWS 區域的。

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

主要 CloudFormation 堆疊的 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

ec2AmiInfo

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 ID。

類型：字串

region

建立映像 AWS 區域的。

類型：字串

version

用來建置映像的 AWS ParallelCluster 版本。

類型：字串

範例

Python

請求

```
$ delete_image(custom-image-id)
```

200 個回應

```
{
  "image": {
    "image_build_status": "DELETE_IN_PROGRESS",
    "image_id": "custom-image-id",
    "region": "us-east-1",
    "version": "3.2.1"
  }
}
```

describeCluster

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

clusterName

叢集的名稱。

類型：字串

必要：是

region

AWS 區域 叢集所在的。

類型：字串

必要：否

回應語法

Note

failureReason 從 3.5.0 AWS ParallelCluster 版failures開始，已變更為。

```
{
  "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",
  "poolName": "string",
  "scheme": "string",
  "healthyNodes": integer,
  "unhealthyNodes": integer
}
}
```

回應內文

clusterName

叢集的名稱。

類型：字串

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (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

clusterConfiguration

url

叢集組態檔案的 URL。

類型：字串

clusterStatus

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

運算機群狀態。

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

叢集上次更新時間的時間戳記。

類型：日期時間

region

建立叢集 AWS 區域的。

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。

failureCode

叢集堆疊處於 CREATE_FAILED 狀態時的失敗碼。

類型：字串

failureReason

當叢集堆疊處於 CREATE_FAILED 狀態時失敗的原因。

類型：字串

head_node

叢集前端節點。

instanceId

Amazon EC2 執行個體 ID。

類型：字串

instanceType

Amazon EC2 執行個體類型。

類型：字串

launchTime

Amazon EC2 執行個體啟動的時間。

類型：日期時間

privateIpAddress

叢集私有 IP 地址。

類型：字串

publicIpAddress

叢集公有 IP 地址。

類型：字串

state

前端節點執行個體狀態。

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

loginNodes

status

登入節點狀態。

類型：字串

有效值：PENDING | FAILED | ACTIVE

address

登入節點地址。

類型：字串

poolName

登入節點集區名稱。

類型：字串

結構描述

登入節點方案。

類型：字串

healthyNodes

運作狀態良好的節點數目。

類型：整數

unhealthyNodes

運作狀態不佳的節點數目。

類型：整數

type

排程器類型。

類型：字串

範例

Python

請求

```
$ describe_cluster(cluster_name_3x)
```

200 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

描述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

clusterName

叢集的名稱。

類型：字串

必要：是

nextToken

下一組結果的字符。

類型：字串

必要：否

nodeType

依節點類型篩選執行個體。

類型：字串

有效值：HeadNode、ComputeNode、LoginNode

必要：否

queueName

依佇列名稱篩選執行個體。

類型：字串

必要：否

region

AWS 區域 叢集所在的。

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",
    "poolName": "string"
  }
]
```

回應內文

執行個體

叢集執行個體的清單。

instanceId

Amazon EC2 執行個體 ID。

類型：字串

instanceType

Amazon EC2 執行個體類型。

類型：字串

launchTime

Amazon EC2 執行個體啟動的時間。

類型：日期時間

nodeType

節點類型。

類型：字串

有效值：HeadNode、ComputeNode、LoginNode

publicIpAddress

叢集公有 IP 地址。

類型：字串

queueName

Amazon EC2 執行個體正在備份節點的佇列名稱。

類型：字串

state

節點 Amazon EC2 執行個體狀態。

類型：字串

有效值：pending | running | shutting-down | terminated | stopping | stopped

nextToken

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求

```
$ describe_cluster_instances(cluster_name_3x)
```

200 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

描述運算機群的狀態。

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

clusterName

叢集的名稱。

類型：字串

必要：是

region

AWS 區域 叢集所在的。

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

lastStatusUpdatedTime

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

```
$ describe_compute_fleet(cluster_name_3x)
```

200 個回應

```
{
  "last_status_updated_time": datetime.datetime(2022, 3, 28, 22, 27, 14,
  tzinfo=tzlocal()),
  "status": "RUNNING"
}
```

describeImage

取得現有映像的詳細資訊。

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 ID。

類型：字串

必要：是

region

建立映像 AWS 區域的。

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

imageConfiguration

url

映像組態檔案的 URL。

類型：字串

region

建立映像 AWS 區域的。

類型：字串

version

用來建置映像的 AWS ParallelCluster 版本。

類型：字串

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (ARN)。

類型：字串

cloudformationStackCreationTime

建立 CloudFormation 堆疊的時間。

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

cloudformationStackStatusReason

CloudFormation 堆疊狀態的原因。

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

建立映像的時間。

類型：日期時間

ec2AmiInfo

amiId

Amazon EC2 AMI ID。

類型：字串

amiName

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 標籤的清單。

金鑰

標籤名稱。

類型：字串

value

標籤值。

類型：字串

imagebuilderImageStatus

ImageBuilder 映像狀態。

類型：字串

有效值：PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING | INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

imagebuilderImageStatusReason

ImageBuilder Image 具有該狀態的原因。

類型：字串

imageBuildLogsArn

映像建置程序日誌的 Amazon Resource Name (ARN)。

類型：字串

範例

Python

請求

```
$ describe_image(custom-image-id)
```

200 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:amzn-s3-demo-bucket',
      "value": 'amzn-s3-demo-bucket'
    },
    {
      "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://amzn-s3-demo-bucket/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://amzn-s3-demo-bucket.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"
}
```

getClusterLogEvents

擷取與日誌串流相關聯的事件。

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

clusterName

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，以 ISO 8601 格式表示。時間戳記等於或晚於此時間的事件不會包含在內。

類型：日期時間

格式：2021-01-01T20:00:00Z

必要：否

limit

傳回的日誌事件數目上限。如果您未指定值，則最大值為回應大小為 1 MB 或最多 10,000 個日誌事件的日誌事件數量。

類型：浮動

必要：否

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 叢集所在的。

類型：字串

必要：否

startFromHead

如果設定為 true，則會先傳回最早的日誌事件。如果值為 false，則會先傳回最新的日誌事件。預設值為 false。

類型：布林值

必要：否

startTime

時間範圍的開始，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

篩選的事件清單。

message

事件訊息。

類型：字串

timestamp

事件時間戳記。

類型：日期時間

nextToken

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

prevToken

可用於擷取上一組結果，或沒有其他結果null的字符。

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)
```

200 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"
  },
  ...
]
```

getClusterStackEvents

擷取與叢集堆疊相關聯的事件。

Note

從 3.6.0 版開始，AWS ParallelCluster 會使用巢狀堆疊來建立與佇列和運算資源相關聯的資源。GetClusterStackEvents API 和 `pcluster get-cluster-stack-events` 命令只會傳回叢集主要堆疊事件。您可以在 CloudFormation 主控台中檢視叢集堆疊事件，包括與佇列和運算資源相關的事件。

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

clusterName

叢集的名稱。

類型：字串

必要：是

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 叢集所在的。

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

篩選的事件清單。

`clientRequestToken`

權杖會傳遞至產生此事件的動作。

類型：字串

`eventId`

此事件的唯一 ID。

類型：字串

`logicalResourceId`

範本中指定資源的邏輯名稱。

類型：字串

`physicalResourceId`

與資源實體執行個體相關聯的名稱或唯一識別符。

類型：字串

`resourceProperties`

用來建立資源的屬性 BLOB。

類型：字串

`resourceStatus`

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

stackId

堆疊執行個體的唯一 ID 名稱。

類型：字串

stackName

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

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求


```
$ get_cluster_stack_events(cluster_name_3x)
```

200 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())
    },
    ...
  ]
}
```

getImageLogEvents

擷取與映像建置相關聯的事件。

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 ID。

類型：字串

必要：是

logStreamName

Logstream 的名稱。

類型：字串

必要：是

endTime

時間範圍的結束，以 ISO 8601 格式表示。時間戳記等於或晚於此時間的事件不會包含在內。

類型：日期時間

格式：2021-01-01T20:00:00Z

必要：否

limit

傳回的日誌事件數目上限。如果您未指定值，則日誌事件的數量上限為 1 MB 的回應大小，最多 10,000 個日誌事件。

類型：浮動

必要：否

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 映像所在的。

類型：字串

必要：否

startFromHead

如果設定為 `true`，請先傳回最早的日誌事件。如果設定為 `false`，請先傳回最新的日誌事件。預設值為 `false`。

類型：布林值

必要：否

startTime

時間範圍的開始，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

篩選的事件清單。

message

事件訊息。

類型：字串

timestamp

事件時間戳記。

類型：日期時間

nextToken

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

prevToken

可用於擷取上一組結果，或沒有其他結果null的字符。

類型：字串

範例

Python

請求

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

200 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"  
  },  
  ...  
]
```

getImageStackEvents

擷取與映像建置的堆疊相關聯的事件。

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 ID。

類型：字串

必要：是

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 映像所在的。

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

篩選的事件清單。

clientRequestToken

權杖會傳遞至產生此事件的動作。

類型：字串

eventId

此事件的唯一 ID。

類型：字串

logicalResourceId

範本中指定資源的邏輯名稱。

類型：字串

physicalResourceId

與資源實體執行個體相關聯的名稱或唯一識別符。

類型：字串

resourceProperties

用來建立資源的屬性 BLOB。

類型：字串

resourceStatus

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

stackId

堆疊執行個體的唯一 ID 名稱。

類型：字串

stackName

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

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求

```
$ get_image_stack_events(image_id)
```

200 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())
  },
  ...
]
```

listClusters

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

clusterStatus

依叢集狀態篩選。預設值是所有叢集。

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

下一組結果的字符。

類型：字串

必要：否

region

叢集 AWS 區域的。

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

主要 CloudFormation 堆疊的 Amazon Resource Name (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

clusterName

叢集的名稱。

類型：字串

clusterStatus

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

建立叢集 AWS 區域的。

類型：字串

version

用來建立叢集的 AWS ParallelCluster 版本。

類型：字串

nextToken

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求

```
$ list_clusters()
```

200 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'
    },
    ...
  ]
}
```

listClusterLogStreams

擷取與叢集相關聯的日誌串流清單。

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

clusterName

叢集的名稱。

類型：字串

必要：是

篩選條件

篩選日誌串流。

可接受的篩選條件為：

- private-dns-name：執行個體私有 DNS 名稱的簡短形式（例如 ip-10-0-0-101）。
- node-type：有效值：HeadNode。

類型：字串陣列是唯一的

格式：Name=a,Values=1 Name=b,Values=2,3

必要：否

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 叢集所在的。

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

logStreams

日誌串流的清單。

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

上次串流事件的時間。lastEventTime 值會根據最終一致性進行更新。它通常從擷取不到一小時就更新，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

日誌串流的 Amazon Resource Name (ARN)。

類型：字串

logStreamName

日誌串流的名稱。

類型：字串

uploadSequenceToken

序列字符。

類型：字串

nextToken

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求

```
$ list_cluster_log_streams(cluster_name_3x)
```


200 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': '####'
    },
    ...
  ]
}
```

listImageLogStreams

擷取與映像相關聯的日誌串流清單。

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 ID。

類型：字串

必要：是

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 映像所在的。

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

logStreams

日誌串流的清單。

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

上次串流事件的時間。lastEventTime 值會根據最終一致性進行更新。它通常從擷取不到一小時就更新，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

日誌串流的 Amazon Resource Name (ARN)。

類型：字串

logStreamName

日誌串流的名稱。

類型：字串

uploadSequenceToken

序列字符。

類型：字串

next_token

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求

```
$ list_image_log_streams(custom-image-id)
```

200 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': '####'
    },
    ...
  ]
}
```

listImages

擷取現有自訂映像的清單。

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

imageStatus

依提供的狀態篩選影像。

類型：字串

有效值：AVAILABLE | PENDING | FAILED

必要：是

nextToken

下一組結果的字符。

類型：字串

必要：否

region

AWS 區域 影像所在的。

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

影像清單。

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (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

ec2AmiInfo

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 ID。

類型：字串

region

建立映像 AWS 區域的。

類型：字串

version

用來建置映像的 AWS ParallelCluster 版本。

類型：字串

nextToken

可用於擷取下一組結果的字符，或如果沒有其他結果則為 null。

類型：字串

範例

Python

請求

```
$ list_images("AVAILABLE")
```

200 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

擷取 AWS ParallelCluster 官方映像的清單。

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/images/official
```



```
{
  "architecture": "string",
  "os": "string",
  "region": "string"
}
```

請求內文

架構

依架構篩選。預設為不篩選。

類型：字串

有效值：x86_64 | arm64

必要：否

OS

依作業系統分佈篩選。預設為不篩選。

類型：字串

有效值：alinux2 | alinux2023 | ubuntu2404 | ubuntu2204 | rhel8 | rhel9

必要：否

region

列出正式映像 AWS 區域的。

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

amild

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

AWS ParallelCluster 版本。

類型：字串

範例

Python

請求

```
$ list_official_images()
```

200 個回應

```
{
  'images': [
    {
      'ami_id': 'ami-015cfef4e0d6306b2',
      'architecture': 'x86_64',
      'name': 'aws-parallelcluster-3.2.1-ubuntu-2204-lts-hvm-x86_64-202202261505 '
      '2022-02-26T15-08-34.759Z',
      'os': 'ubuntu2204',
      'version': '3.2.1'
    },
    ...
  ]
}
```

updateCluster

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

clusterConfiguration

叢集組態做為 YAML 文件。

必要：是

clusterName

叢集的名稱。

類型：字串

必要：是

dryrun

如果設定為 true，則只執行請求驗證，而不建立任何資源。使用此參數來驗證叢集組態和更新需求。預設值為 false。

類型：布林值

必要：否

forceUpdate

如果設定為 true，請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

AWS 區域 叢集所在的。

類型：字串

必要：否

suppressValidators

識別要隱藏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

導致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

叢集更新的變更集。

currentValue

要更新的參數目前值。

類型：字串

參數

要更新的參數。

類型：字串

requestedValue

要更新的參數請求值。

類型：字串

叢集

cloudformationStackArn

主要 CloudFormation 堆疊的 Amazon Resource Name (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

clusterName

叢集的名稱。

類型：字串

clusterStatus

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

建立叢集 AWS 區域的。

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

validationMessages

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

驗證器的 ID。

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)
```

200 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

更新叢集運算機群的狀態。

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

clusterName

叢集的名稱。

類型：字串

必要：是

status

運算機群狀態。

類型：字串

有效值：START_REQUESTED | STOP_REQUESTED | ENABLED | DISABLED

必要：是

region

AWS 區域 叢集所在的。

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

運算機群狀態。

類型：字串

有效值：START_REQUESTED | STARTING | RUNNING | PROTECTED | STOP_REQUESTED
| STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

lastStatusUpdatedTime

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")
```

200 個回應

```
{
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
    tzinfo=tzlocal()),
  'status': 'START_REQUESTED'
}
```

AWS ParallelCluster Python 程式庫 API

從 3.5.0 AWS ParallelCluster 版開始，您可以使用 AWS ParallelCluster Python AWS ParallelCluster 程式庫存取。您可以在 `pcluster` 環境中或在 AWS Lambda 執行時間內存取程式 AWS

ParallelCluster 庫。了解如何使用 AWS ParallelCluster Python AWS ParallelCluster 程式庫存取 API。AWS ParallelCluster Python 程式庫提供與 AWS ParallelCluster API 相同的功能。

AWS ParallelCluster Python 程式庫操作和參數會在轉換為snake_case不含大寫字母的 API 參數時鏡像這些參數。

主題

- [AWS ParallelCluster Python 程式庫授權](#)
- [安裝 AWS ParallelCluster Python 程式庫](#)
- [叢集 API 操作](#)
- [運算機群 API 操作](#)
- [叢集和堆疊日誌操作](#)
- [映像 API 操作](#)
- [映像和堆疊日誌操作](#)
- [範例](#)
- [AWS Lambda 適用於 AWS ParallelCluster Python 程式庫](#)

AWS ParallelCluster Python 程式庫授權

使用任何對 boto3 有效的標準方式來指定登入資料。如需詳細資訊，請參閱 [boto3 文件](#)。

安裝 AWS ParallelCluster Python 程式庫

1. 遵循 [中的指示](#) 安裝 pcluster CLI 3.5.0 版或更新版本。 [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 操作

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

下一組結果的字符。

cluster_status

依叢集狀態篩選。預設為列出所有叢集。

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)
```

在指定區域中建立叢集。

參數：

cluster_name (必要)

叢集名稱。

cluster_configuration (必要)

叢集組態做為 Python 資料類型。

region

叢集 AWS 區域。

suppress_validators

識別要隱藏的一或多個叢集組態驗證程式。

格式：(ALL | type:[A-Za-z0-9]+)

validation_failure_level

導致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

執行請求驗證，而不建立任何資源。您可以使用它來驗證叢集組態。預設值為 False。

rollback_on_failure

如果設定為 True，會在失敗時 AWS ParallelCluster 自動啟動叢集堆疊復原。預設值為 True。

wait

如果設定為 True，AWS ParallelCluster 會等待操作完成。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，會等待操作完成。預設值為 False。

describe_cluster

```
describe_cluster(cluster_name, region)
```

取得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)

叢集組態做為 Python 資料類型。

suppress_validators

識別要隱藏的一或多個叢集組態驗證程式。

格式：(ALL | type:[A-Za-z0-9]+)

validation_failure_level

導致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

執行請求驗證，而不建立或更新任何資源。您可以使用它來驗證叢集組態。預設值為 False。

force_update

如果設定為 True，會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，會等待操作完成。預設值為 False。

運算機群 API 操作

主題

- [describe_compute_fleet](#)
- [update_compute_fleet](#)
- [delete_cluster_instances](#)
- [describe_cluster_instances](#)

describe_compute_fleet

```
describe_compute_fleet(cluster_name, region)
```

描述指定叢集的叢集運算機群狀態。

參數：

cluster_name (必要)

叢集名稱。

region

描述部署到指定叢集的運算機群狀態 AWS 區域。

update_compute_fleet

```
update_compute_fleet(cluster_name, status, region)
```

更新叢集運算機群的狀態。

參數：

cluster_name (必要)

叢集名稱。

status (必要)

要更新的狀態。

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)
```

啟動強制終止所有叢集運算節點。此動作不支援 AWS Batch 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，則在 `cluster_name` 找不到具有指定的叢集時強制刪除。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

下一組結果的字符。

node_type

依篩選執行個體 `node_type`。

有效值：HeadNode | ComputeNode

queue_name

依佇列名稱篩選執行個體。

叢集和堆疊日誌操作

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)
```

列出指定叢集的日誌串流。

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'

已接受的篩選條件：

code-dns-name

執行個體私有 DNS 名稱的簡短形式，例如 ip-10-0-0-101。

node-type

節點類型。

有效值：HeadNode

next_token

下一組結果的字符。

get_cluster_log_events

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,  
start_from_head, limit, start_time, end_time)
```

取得指定叢集和日誌串流的日誌事件。

參數：

cluster_name (必要)

叢集名稱。

log_stream_name (必要)

日誌串流名稱。

region

叢集 AWS 區域。

next_token

下一組結果的字符。

start_from_head

如果設定為 True，會先 AWS ParallelCluster 傳回最早的日誌事件。如果設定為 False，它會先傳回最新的日誌事件。預設值為 False。

limit

傳回の日誌事件數目上限。如果您未指定值，則最大數量為回應大小為 1 MB 的日誌數量，最多可達 10,000 個日誌事件。

start_time

日誌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

日誌事件的時間範圍結束，以 ISO 8601 格式表示；例如 '2021-01-01T20:00:00Z'。時間戳記等於或晚於此時間的事件不會包含在內。

get_cluster_stack_events

```
get_cluster_stack_events(cluster_name, region, next_token)
```

取得指定叢集的堆疊事件。

參數：

cluster_name (必要)

叢集名稱。

region

叢集 AWS 區域。

next_token

下一組結果的字符。

映像 API 操作

主題

- [list_images](#)
- [build_image](#)
- [delete_image](#)
- [describe_image](#)

list_images

```
list_images(image_status, region, next_token)
```

擷取現有映像的清單。

參數：

image_status (必要)

依影像狀態篩選。

有效值：AVAILABLE | PENDING | FAILED

region

列出在指定 中建置的映像 AWS 區域。

next_token

下一組結果的字符。

build_image

```
build_image(image_configuration, image_id, suppress_validators,
            validation_failure_level, dry_run, rollback_on_failure, region)
```

在指定區域中建立自訂 AWS ParallelCluster 映像。

參數：

image_configuration (必要)

影像組態為 Python 資料。

image_id (必要)

映像 ID。

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

導致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，會 AWS ParallelCluster 執行請求驗證，而不建立任何資源。您可以使用它來驗證映像組態。預設值為 False。

rollback_on_failure

如果設定為 True，會在失敗時 AWS ParallelCluster 自動啟動映像堆疊轉返。預設值為 False。

region

影像 AWS 區域。

delete_image

```
delete_image(image_id, region, force)
```

刪除指定區域中的映像。

參數：

image_id (必要)

映像 ID。

region

影像 AWS 區域。

force

如果設定為 True，當執行個體使用 AMI 或共用 AMI 時，AWS ParallelCluster 強制刪除。預設值為 False。

describe_image

```
describe_image(image_id, region)
```

取得現有映像的詳細資訊。

參數：

image_id (必要)

映像 ID。

region

影像 AWS 區域。

映像和堆疊日誌操作

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)
```

列出映像的日誌串流。

參數：

image_id (必要)

映像 ID。

region

影像 AWS 區域。

next_token

下一組結果的字符。

get_image_log_events

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head, limit, start_time, end_time)
```

取得指定映像和日誌串流的日誌事件。

參數：

image_id (必要)

映像 ID。

log_stream_name (必要)

日誌串流名稱。

region

影像 AWS 區域。

next_token

下一組結果的字符。

start_from_head

如果設定為 True，會先 AWS ParallelCluster 傳回最早的日誌事件。如果設定為 False，它會先傳回最新的日誌事件。預設值為 False。

limit

傳回的日誌事件數目上限。如果您未指定值，則最大數量為回應大小為 1 MB 的日誌數量，最多可達 10,000 個日誌事件。

start_time

日誌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

日誌事件的時間範圍結束，以 ISO 8601 格式表示；例如 '2021-01-01T20:00:00Z'。時間戳記等於或晚於此時間的事件不會包含在內。

get_image_stack_events

```
get_image_stack_events(image_id, region, next_token)
```

取得指定映像的堆疊事件。

參數：

image_id (必要)

映像 ID。

region

影像 AWS 區域。

next_token

下一組結果的字符。

list_official_images

```
list_official_images(region,os, architecture)
```

擷取官方 AWS ParallelCluster 映像的清單。

參數：

region

影像 AWS 區域。

os

依作業系統分佈篩選。預設為不篩選。

architecture

依架構篩選。預設為不篩選。

範例

主題

- [建立 叢集](#)

建立 叢集

當您執行下列範例指令碼時，使用存放在您環境中的指定輸入，您可以建立叢集。叢集組態會根據[叢集組態文件](#)建立為 Python 資料類型。

```
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': '3.15.0'}}

```

AWS Lambda 適用於 AWS ParallelCluster Python 程式庫

您可以部署 Lambda 層和執行時間來存取 AWS ParallelCluster Python 程式庫。我們會託管 zip AWS ParallelCluster 檔案，您可以輸入 zip 檔案的連結，如下列步驟所述。Lambda 使用 zip 檔案來準備執行期環境，以支援對 Python 程式庫的存取。AWS ParallelCluster Python 程式庫以 3.5.0 AWS ParallelCluster 版新增。您只能將 程式庫用於 3.5.0 版和更新版本。

託管的 zip 檔案 URL 格式為：<s3://aws-region-id-aws-parallelcluster/parallelcluster/3.15.0/layers/aws-parallelcluster/lambda-layer.zip>。（將 **3.15.0** 取代為您想要在下列步驟中使用的 AWS ParallelCluster 版本。）

開始使用 存取 AWS ParallelCluster Python 程式庫 AWS Lambda

建立 Lambda 層

1. 登入 AWS 管理主控台 並導覽至 AWS Lambda 主控台。
2. 在導覽窗格中，選取圖層，然後選取建立圖層。

3. 輸入 layer 的名稱，然後選取從 Amazon S3 上傳檔案。
4. 輸入 zip 檔案的 URL：`s3://aws-region-id-aws-parallelcluster/parallelcluster/3.15.0/layers/aws-parallelcluster/lambda-layer.zip`。
5. 針對相容的架構，選擇 x86_64 架構。
6. 針對相容的執行時間，選擇 Python 3.12 執行時間。
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，然後選取建立函數。
2. 輸入函數的名稱。
3. 針對執行時間，選擇 Python 3.12 執行時間。
4. 針對架構，選擇 x86_64 架構。
5. 選擇建立函數。
6. 建立函數之後，請選擇圖層，然後選取新增圖層。
7. 選取自訂圖層，然後選擇您在先前步驟中建立的圖層。
8. 選擇 layer 版本。
9. 選擇新增。
10. 您的 Lambda 需要許可才能管理使用建立的叢集 AWS ParallelCluster。使用中列出的許可建立 Lambda 角色 [基本 AWS ParallelCluster pcluster 使用者政策](#)。

您現在可以 AWS ParallelCluster 從 Python 程式庫存取，如中所述 [AWS ParallelCluster Python 程式庫 API](#)。

如何使用的教學課程 AWS ParallelCluster

下列教學課程說明如何開始使用第 3 AWS ParallelCluster 版，並提供一些常見任務的最佳實務指引。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

主題

- [在上執行您的第一個任務 AWS ParallelCluster](#)
- [建置自訂 AWS ParallelCluster AMI](#)
- [整合 Active Directory](#)
- [使用 AWS KMS 金鑰設定共用儲存加密](#)
- [在多個佇列模式叢集中執行任務](#)
- [使用 AWS ParallelCluster API](#)
- [使用 Slurm 會計建立叢集](#)
- [使用外部 Slurmdbd 會計建立叢集](#)
- [還原至先前的 AWS Systems Manager 文件版本](#)
- [使用 建立叢集 CloudFormation](#)
- [使用 Terraform 部署 ParallelCluster API](#)
- [使用 Terraform 建立叢集](#)
- [使用 Terraform 建立自訂 AMI](#)
- [AWS ParallelCluster UI 與 Identity Center 整合](#)
- [使用 Pyxis 執行容器化任務](#)
- [使用啟用 EFA 的 FSx Lustre 建立叢集](#)
- [使用 p6e-gb200 執行個體支援 NVIDIA-Imex](#)
- [使用啟動範本覆寫自訂運算節點網路介面](#)

在上執行您的第一個任務 AWS ParallelCluster

本教學課程會逐步引導您在 上執行第一個 Hello World 任務 AWS ParallelCluster

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

先決條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有 [Amazon EC2 金鑰對](#)。
- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的[許可](#)。

確認安裝

首先，我們會驗證 AWS ParallelCluster 是否正確安裝和設定，包括 Node.js 相依性。

```
$ node --version
v16.8.0
$ pcluster version
{
  "version": "3.15.0"
}
```

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

大多數 pcluster 命令都必須指定 AWS 區域要使用的。如果未在 `AWS_DEFAULT_REGION` 環境變數中指定，或 `~/.aws/config` 檔案 [default] 區段中的 `region` 設定，則必須在 pcluster 命令列中提供 `--region` 參數。

如果輸出提供您有關組態的訊息，您需要執行下列動作來設定 AWS ParallelCluster：

```
$ pcluster configure --config hello-world.yaml
```

如果 `pcluster create-cluster` 命令成功，您會看到類似以下的輸出：

```
{
  "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控叢集的建立：

```
$ pcluster describe-cluster --cluster-name hello-world
```

在建立叢集時 `clusterStatus` 報告「CREATE_IN_PROGRESS」。成功建立叢集時，會 `clusterStatus` 轉換為「CREATE_COMPLETE」。輸出也提供我們前端節點 `privateIpAddress` 的 `publicIpAddress` 和。

登入您的頭部節點

使用您的 OpenSSH pem 檔案登入您的頭部節點。

```
$ 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]
```

輸出顯示我們的叢集中有一個佇列，最多十個節點。

使用 Slurm 執行您的第一個任務

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中啟動。您可以使用 `sinfo` 命令監控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
```

輸出顯示任務已提交至 `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「out」檔案。我們可以看到任務的輸出：

```
$ 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。

若要進一步了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：

- 建置 AWS ParallelCluster 已啟用的自訂 AMI，其中包含您的軟體，如中所述 [建置自訂 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，並將已安裝的軟體存放在指定的掛載位置。
- 使用 [自訂引導操作](#) 來自動化叢集中每個節點的引導程序。

建置自訂 AWS ParallelCluster AMI

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱 [AWS 使用的服務 AWS ParallelCluster](#)。

Important

如果您建置自訂 AMI，則必須在每個新 AWS ParallelCluster 版本中重複用來建立自訂 AMI 的步驟。

在進一步閱讀之前，我們建議您先檢閱 [自訂引導操作](#) 區段。判斷您要進行的修改是否可以編寫指令碼，並支援未來的 AWS ParallelCluster 版本。

雖然通常建立自訂 AMI 並不理想，但在某些情況下 AWS ParallelCluster，需要為建置自訂 AMI。本教學課程說明如何為這些案例建置自訂 AMI。

先決條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有 [Amazon EC2 金鑰對](#)。
- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 和建置映像所需的 [許可](#)。

如何自訂 AWS ParallelCluster AMI

有兩種方式可建置自訂 AWS ParallelCluster AMI。這兩種方法之一是使用 CLI AWS ParallelCluster 建立新的 AMI。另一種方法需要您手動修改，以建置可在下使用的新 AMI AWS 帳戶。

建置自訂 AWS ParallelCluster AMI

如果您有自訂 AMI 和軟體，您可以在其 AWS ParallelCluster 上套用所需的變更。AWS ParallelCluster 依賴 EC2 Image Builder 服務來建置自訂 AMIs。如需詳細資訊，請參閱[映像建置器使用者指南](#)。

重點：

- 程序大約需要 1 小時。如果在建置時要安裝其他 [Build /Components](#)，則此時間可能會有所不同。
- AMI 會標記主要元件的版本。這些包括核心、排程器和 [EFA](#) 驅動程式。元件版本的子集也會在 AMI 描述中報告。
- 從 AWS ParallelCluster 3.0.0 開始，一組新的 CLI 命令可用來管理映像的生命週期。其中包括 [build-image](#)、[list-images](#)、[describe-image](#) 和 [delete-image](#)。
- 此方法可重複。您可以重新執行它來保持 AMIs 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

如果您在 AWS 中國分割區中使用此方法，您可能會遇到網路錯誤。例如，當您從 GitHub 或作業系統儲存庫下載套件時，您可能會從 `pcluster build-image` 命令看到這些錯誤。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 [修改 AWS ParallelCluster AMI](#) 的方法。
2. 在另一個分割區和區域中建置映像，例如 `us-east-1`，然後存放/還原它以將其移至中國區域。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 登入資料，讓 AWS ParallelCluster 用戶端可以代表您呼叫 AWS API 操作。如需必要許可的清單，請參閱 [AWS Identity and Access Management](#) 中的 [許可 AWS ParallelCluster](#)。

2. 建立基本建置映像組態檔案。若要這樣做，請指定 [InstanceType](#) 要用來建置映像的 和 [ParentImage](#)。這些是用來建立 AMI 的起點。如需選用建置參數的詳細資訊，請參閱 [映像組態](#)。

```
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 AWS ParallelCluster AMI 開始建置 AMI。

```
$ 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": "3.15.0"
  }
}
```

Warning

`pcluster build-image` 使用預設 VPC。如果您使用 AWS Control Tower 或 AWS 登陸區域刪除預設 VPC，則必須在映像組態檔案中指定子網路 ID。如需詳細資訊，請參閱 [SubnetId](#)。

如需其他參數的清單，請參閱 [pcluster build-image](#) 命令參考頁面。上述命令的結果如下：

- 根據映像組態建立 CloudFormation 堆疊。堆疊包含建置所需的所有 EC2 Image Builder 資源。
- 建立的資源包括自訂映像建置器 AWS ParallelCluster 元件可新增至的官方映像建置器元件。如需詳細資訊，請參閱 EC2 [Image Builder 使用者指南中的使用映像建置器建立自訂元件](#)。
- EC2 Image Builder 會啟動建置執行個體、套用 AWS ParallelCluster 技術指南、安裝 AWS ParallelCluster 軟體堆疊，以及執行必要的組態任務。AWS ParallelCluster 技術指南用於建置和引導 AWS ParallelCluster。
- 執行個體會停止，並從中建立新的 AMI。

- 另一個執行個體會從新建立的 AMI 啟動。在測試階段，EC2 Image Builder 會執行 Image Builder 元件中定義的測試。
 - 如果建置成功，則會刪除堆疊。如果建置失敗，堆疊會保留並可供檢查。
4. 您可以執行下列命令來監控建置程序的狀態。建置完成後，您可以執行它來擷取回應中提供的 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/3.15.0/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": "3.15.0",
  "cloudformationStackTags": [
    {
      "value": "3.15.0",
      "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/3.15.0/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": "3.15.0"
}
```

5. 若要建立叢集，請在叢集組態的 [CustomAmi](#) 欄位中輸入 AMI ID。

故障診斷和監控 AMI 建立程序

影像建立會在大約一小時的時間內完成。您可以執行 [pcluster describe-image](#) 命令或日誌擷取命令來監控程序。

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

此 [build-image](#) 命令會使用建置映像所需的所有 Amazon EC2 資源建立 CloudFormation 堆疊，並啟動 EC2 Image Builder 程序。

執行 [build-image](#) 命令後，您可以使用 擷取 CloudFormation 堆疊事件 [pcluster get-image-stack-events](#)。您可以使用 `--query` 參數篩選結果，以查看最新的事件。如需詳細資訊，請參閱 AWS Command Line Interface 《使用者指南》中的 [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/3.15.0/1",
  "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/3.15.0\\\",\\\"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"
}
```

約 15 分鐘後，堆疊事件會出現在與映像建置器建立相關的日誌事件項目中。您現在可以使用 [和 `pcluster get-image-log-events` 命令列出映像日誌串流](#)[pcluster list-image-log-streams](#) 並監控映像建置器步驟。

```
$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
--query 'logStreams[*].logStreamName'

"3.15.0/1"
]

$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
--log-stream-name 3.15.0/1 --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-ij90-1234abcd5678/3.15.0/1",
    "timestamp": "2022-04-05T22:13:26.741Z"
  },
  {
    "message": "TOE has completed execution successfully",
    "timestamp": "2022-04-05T22:13:26.819Z"
  }
]
}

```

繼續檢查 [describe-image](#) 命令，直到您看到 BUILD_COMPLETE 狀態為止。

```

$ 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/3.15.0/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": "3.15.0"
}

```

```
}
```

如果您需要對自訂 AMI 建立問題進行疑難排解，請建立映像日誌的封存，如下列步驟所述。

視 `--output` 參數而定，可以將日誌存檔在 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"  
}
```

封存包含與映像建置器程序和 CloudFormation 堆疊事件相關的 CloudWatch Logs 串流。命令可能需要幾分鐘的時間才能執行。

管理自訂 AMIs

從 AWS ParallelCluster 3.0.0 開始，在 CLI 中新增一組新的命令，以建置、監控和管理映像生命週期。如需 命令的詳細資訊，請參閱 [pcluster 命令](#)。

修改 AWS ParallelCluster AMI

此方法包含透過在官方 AWS ParallelCluster AMI 上新增自訂來修改官方 AMI。AWS ParallelCluster AMIs 會以新版本更新。這些 AMIs 具有安裝和設定時，AWS ParallelCluster 運作所需的所有元件。您可以從其中一個開始，做為您的基礎。

重點：

- 此方法比 [build-image](#) 命令更快。不過，這是手動程序，無法自動重複。
- 透過此方法，您無法存取可透過 CLI 取得的日誌擷取和映像生命週期管理命令。

步驟：

New Amazon EC2 console

1. 尋找對應至您使用之特定 AWS 區域的 AMI。若要尋找，請使用 [pcluster list-official-images](#) 命令搭配 `--region` 參數，以選取特定 AWS 區域和 `--os` 和 `--architecture` 參數，以您想要使用的作業系統和架構來篩選所需的 AMI。從輸出擷取 Amazon EC2 映像 ID。
2. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
3. 在導覽窗格中，選擇影像，然後選擇 AMIs。搜尋擷取的 EC2 映像 ID，選取 AMI，然後從 AMI 選擇啟動執行個體。
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. 執行下列命令來準備您的執行個體以建立 AMI。

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. 從主控台中，選擇執行個體狀態和停止執行個體。

導覽至執行個體、選擇新執行個體、選取執行個體狀態和停止執行個體。

10. 使用 Amazon EC2 主控台或 AWS CLI [create-image](#) 從執行個體建立新的 AMI。

從 Amazon EC2 主控台

- a. 在導覽窗格中，選擇 Instances (執行個體)。
 - b. 選擇您建立和修改的執行個體。
 - c. 在動作中，選擇映像，然後選擇建立映像。
 - d. 選擇 Create Image (建立映像)。
11. 在叢集組態的 [CustomAmi](#) 欄位中輸入新的 AMI ID，然後建立叢集。

Old Amazon EC2 console

1. 尋找對應至您使用之特定 AWS 區域的 AWS ParallelCluster AMI。若要尋找它，您可以使用 [pcluster list-official-images](#) 命令搭配 `--region` 參數，選取特定 AWS 區域和 `--os` 和 `--architecture` 參數，以您想要使用的作業系統和架構來篩選所需的 AMI。您可以從輸出擷取 Amazon EC2 Image ID。

2. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/ec2/>：// 開啟 Amazon EC2 主控台。
3. 在導覽窗格中，選擇影像，然後選擇 AMIs。設定公用映像的篩選條件，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您的執行個體。
5. 選擇啟動、選取您的金鑰對和啟動執行個體。
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，請導覽至執行個體，選取新的執行個體並連線。
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. 從 Amazon EC2 主控台，在導覽窗格中選擇執行個體，選擇您的新執行個體，然後選擇動作、執行個體狀態和停止。
10. 使用 Amazon EC2 主控台或 AWS CLI [create-image](#) 從執行個體建立新的 AMI。

從 Amazon EC2 主控台

- a. 在導覽窗格中，選擇 Instances (執行個體)。
 - b. 選擇您建立和修改的執行個體。
 - c. 在動作中，選擇映像，然後選擇建立映像。
 - d. 選擇 Create Image (建立映像)。
11. 在叢集組態的 [CustomAmi](#) 欄位中輸入新的 AMI ID，然後建立叢集。

整合 Active Directory

在本教學課程中，您會建立多個使用者環境。此環境包含與 (Active Directory) 整合 AWS ParallelCluster AWS Managed Microsoft AD 的 `corp.example.com`。您可以設定 Admin 使用者來管理目錄、ReadOnly 使用者來讀取目錄，以及 `user000` 使用者來登入叢集。您可以使用自動化路徑或手動路徑來建立聯網資源、Active Directory (AD)，以及您用來設定 AD 的 Amazon EC2 執行個體。無論您的路徑為何，您建立的基礎設施都會預先設定為 AWS ParallelCluster 使用下列其中一種方法進行整合：

- 具有憑證驗證的 LDAPS (建議為最安全的選項)

- 沒有憑證驗證的 LDAPS
- LDAP

LDAP 本身不提供加密。為了確保安全傳輸潛在敏感資訊，強烈建議您將 LDAPS (LDAP over TLS/SSL) 用於與 ADs 整合的叢集。如需詳細資訊，請參閱《Directory Service 管理指南》中的[使用 啟用伺服器端 LDAPS AWS Managed Microsoft AD](#)。

建立這些資源之後，請繼續設定和建立與 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。如需您在本教學課程中建立組態的詳細資訊，請參閱[多個使用者存取叢集和 DirectoryService 組態區段](#)。

本教學課程說明如何建立支援多個使用者存取叢集的環境。本教學課程不涵蓋您建立和使用 Directory Service AD 的方式。您在本教學 AWS Managed Microsoft AD 課程中設定所採取的步驟僅供測試之用。它們不會用來取代官方文件和最佳實務，請參閱《Directory Service 管理指南》中的[AWS Managed Microsoft AD](#)和[Simple AD](#)。

Note

目錄使用者密碼會根據目錄密碼政策屬性定義過期。若要使用 [重設目錄密碼 AWS ParallelCluster](#)，請參閱[如何重設使用者密碼和過期的密碼](#)。

Note

由於網域控制站變更和目錄維護，目錄網域控制站 IP 地址可能會變更。如果您選擇自動快速建立方法來建立目錄基礎設施，則必須在目錄 IP 地址變更時，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 地址不會自動與負載平衡器對齊。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

先決條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有 [Amazon EC2 金鑰對](#)。

- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的[許可](#)。

當您完成教學課程時，請以您自己的名稱 *region-id* 和 ID 取代 *d-abcdef01234567890*，*inputs highlighted in red* 例如 和 。IDs 將 取代 *0123456789012* 為您的 AWS 帳戶 數字。

建立 AD 基礎設施

選擇自動化索引標籤，使用 CloudFormation 快速建立範本建立 Active Directory (AD) 基礎設施。

選擇手動索引標籤以手動建立 AD 基礎設施。

自動化

1. 登入 AWS 管理主控台。
2. 開啟 [CloudFormation Quick Create \(region us-east-1\)](#)，以在 CloudFormation 主控台中建立下列資源：
 - 如果未指定 VPC，則具有兩個子網路和路由以進行公開存取的 VPC。
 - AWS Managed Microsoft AD。
 - 加入 AD 的 Amazon EC2 執行個體，可用來管理目錄。
3. 在快速建立堆疊頁面參數區段中，輸入下列參數的密碼：
 - AdminPassword
 - ReadOnlyPassword
 - UserPassword

請記下密碼。您稍後在本教學課程中使用它們。
4. 針對 DomainName，輸入 **corp.example.com**
5. 針對 Keypair，輸入 Amazon EC2 金鑰對的名稱。
6. 勾選方塊以確認頁面底部的每個存取功能。
7. 選擇建立堆疊。
8. 在 CloudFormation 堆疊達到 CREATE_COMPLETE 狀態後，選擇堆疊的輸出索引標籤。請記下輸出資源名稱和 IDs，因為您需要在後續步驟中使用它們。輸出提供建立叢集所需的資訊。

CloudFormation > Stacks > PclusterAD-1234abc

Stacks (3) View nested < 1 >

PclusterAD
2022-05-18 10:27:24 UTC-0700
CREATE_COMPLETE

PclusterAD-abcd123

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

Outputs (10)

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 [\(選用\) 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 [\(選用\) 管理 AD 使用者和群組](#) 或 [繼續建立叢集](#)。

手動

為目錄服務建立 VPC，其中兩個子網路位於不同的可用區域和 AWS Managed Microsoft AD。

建立 AD

Note

- 目錄和網域名稱為 `corp.example.com`。簡短名稱為 CORP。
- 變更指令碼中的 Admin 密碼。
- Active Directory (AD) 至少需要 15 分鐘才能建立。

使用下列 Python 指令碼在本機建立 VPC、子網路 and AD 資源 AWS 區域。將此檔案儲存為 `ad.py` 並執行。

```
import boto3
import time
from pprint import pprint

vpc_name = "PclusterVPC"
ad_domain = "corp.example.com"
admin_password = "asdfASDF1234"

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 Python 指令碼的範例輸出。

```
{
  "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 IDs。您可以在後續步驟中使用它們。

指令碼完成後，請繼續下一個步驟。

建立 Amazon EC2 執行個體

New Amazon EC2 console

1. 登入 AWS 管理主控台。
2. 如果您沒有連接步驟 4 中所列政策的角色，請開啟位於 <https://console.aws.amazon.com/iam/> 的 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執行以建立 AD 之指令碼輸出的 AWS 區域 ID、帳戶 ID 和目錄 ID。

ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. 建立已連接下列政策的 IAM 角色。
 - AWS 受管政策 [AmazonSSMManagedInstanceCore](#)
 - AWS 受管政策 [AmazonSSMDirectoryServiceAccess](#)
 - 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，選取最近的 Amazon Linux 2 AMI。
8. 針對執行個體類型，選擇 t2.micro。
9. 針對金鑰對，選擇金鑰對。
10. 針對網路設定，選擇編輯。
11. 針對 VPC，選取目錄 VPC。
12. 向下捲動並選取進階詳細資訊。

13. 在進階詳細資訊的網域聯結目錄中，選擇 **corp.example.com**。
14. 針對 IAM 執行個體描述檔，選擇您在步驟 1 中建立的角色，或連接步驟 4 中列出政策的角色。
15. 在摘要中，選擇啟動執行個體。
16. 請記下執行個體 ID (例如 i-1234567890abcdef0)，並等待執行個體完成啟動。
17. 執行個體啟動後，請繼續下一個步驟。

Old Amazon EC2 console

1. 登入 AWS 管理主控台。
2. 如果您沒有連接步驟 4 中所列政策的角色，請開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。否則，請跳至步驟 5。
3. 建立 ResetUserPassword 政策。將紅色反白內容取代為您執行的指令碼輸出中的 AWS 區域 ID、AWS 帳戶 ID 和目錄 ID，以建立 Active Directory (AD)。

ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. 建立已連接下列政策的 IAM 角色。
 - AWS 受管政策 [AmazonSSMManagedInstanceCore](#)
 - AWS 受管政策 [AmazonSSMDirectoryServiceAccess](#)
 - 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，選取最近的 Amazon Linux 2 AMI。
8. 對於 Instance type (執行個體類型)，選擇 t2.micro。
9. 針對金鑰對，選擇金鑰對。
10. 在網路設定中，選擇編輯。
11. 在網路設定 VPC 中，選取目錄 VPC。
12. 向下捲動並選取進階詳細資訊。
13. 在進階詳細資訊的網域聯結目錄中，選擇 **corp.example.com**。
14. 在進階詳細資訊、執行個體描述檔中，選擇您在步驟 1 中建立的角色，或使用連接步驟 4 中所列政策的角色。
15. 在摘要中，選擇啟動執行個體。
16. 請記下執行個體 ID (例如 i-1234567890abcdef0)，並等待執行個體完成啟動。
17. 執行個體啟動後，請繼續下一個步驟。

將執行個體加入 AD

1. 連線至您的執行個體，並以身分加入 AD 領域 **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. 安裝必要的軟體並加入領域。

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-
common-tools krb5-workstation openldap-clients policycoreutils-python
```

3. 將管理員密碼取代為 **admin** 您的密碼。

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com
Password for Admin:
```

如果上述成功，您就會加入領域，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，您會使用您在上一個步驟中安裝的 [adcli](#) 和 [openldap-clients](#) 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 指令碼的輸出。

```
$ 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"
```

根據預設，當您使用建立使用者時 `adcli`，使用者會停用。

3. 從本機電腦重設和啟用使用者密碼：

登出 Amazon EC2 執行個體。

Note

- `ro-p@ssw0rd` 是從 `ReadOnlyUser` 擷取的密碼 AWS Secrets Manager。
- `user-p@ssw0rd` 是在您將 (ssh) 連線到叢集時提供的叢集使用者密碼。

`directory-id` 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 Secrets Manager 秘密。

現在您已建立 `ReadOnlyUser` 並設定密碼，請將它存放在 AWS ParallelCluster 用於驗證登入的秘密中。

使用 Secrets Manager 建立新的秘密，將的密碼保留 `ReadOnlyUser` 為值。秘密值格式必須是純文字（而非 JSON 格式）。記下秘密 ARN 以供未來步驟使用。

```
$ 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 IDs。您稍後會在步驟中使用它們。

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" | openssl req -x509 -sha256 -nodes -
newkey rsa:2048 -keyout $PRIVATE_KEY -days 365 -out $CERTIFICATE
```

2. 將憑證儲存到 Secrets Manager，以便稍後可從叢集內擷取。

```
$ 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建立的 IAM 角色，以將 Amazon EC2 執行個體加入 AD 網域。

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 Active Directory 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": "Z2IFOLAFXWL04F",  
      "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. 建立目標為 Active Directory 端點的目標群組。

```

$ 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 Active Directory (AD) 端點註冊到目標群組。

```

$ 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，以在叢集 VPC 中探索網域。

```

$ 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. 使用 `recordset-change.json` 下列內容建立名為 `corp.example.com` 的檔案。 `HostedZoneId` 是負載平衡器的正式託管區域 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. 將記錄集變更提交至託管區域，這次使用託管區域 ID。

```

$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXSJUB \
--change-batch file://recordset-change.json
{
  "ChangeInfo": {
    "Id": "/change/C0137926I56R3GC7XW2Y",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
  }
}

```

12. 使用 `policy.json` 下列內容建立政策文件。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-
east-1:123456789012:secret:example-cert-abc123"
      ],
      "Effect": "Allow"
    }
  ]
}

```

```
}
```

13. 使用 `policy.json` 下列內容建立名為 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. 繼續遵循 [\(選用\) 管理 AD 使用者和群組](#) 或 中的步驟 [建立叢集](#)。

(選用) 管理 AD 使用者和群組

在此步驟中，您可以從加入 Active Directory (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，請重新啟動並登入在自動化過程中建立的 AD 聯結執行個體。

如果您遵循手動路徑，請重新啟動並登入您在先前步驟中建立並加入 AD 的執行個體。

在這些步驟中，您會使用在上述步驟中安裝在執行個體中的 [adcli](#) 和 [openldap-clients](#) 工具。

登入已加入 AD 網域的 Amazon EC2 執行個體

1. 從 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。執行個體狀態可能已停止。
2. 如果執行個體狀態為停止，請選擇執行個體狀態，然後選擇啟動執行個體。
3. 狀態檢查通過後，選取執行個體，然後選擇將 和 SSH 連接至執行個體。

在登入加入 AD 的 Amazon EC2 Amazon 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。您可以在 `ldapsearch` 命令中包含 AD Admin 密碼。

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"
```

描述具有名稱模式的所有使用者。

```
$ 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"
```

建立叢集

如果您尚未結束 Amazon EC2 執行個體，請現在執行此操作。

環境設定為建立叢集，可根據 Active Directory (AD) 驗證使用者。

建立簡單的叢集組態，並提供與連接到 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組態，並將其複製到名為 `ldaps_config.yaml`、`ldaps_nocert_config.yaml` 或的檔案 `ldap_config.yaml`。

我們建議您使用憑證驗證選擇 LDAPS 組態。如果您選擇此組態，您還必須將引導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，如組態檔案中所示。

具有憑證驗證組態的 LDAPS (建議)

Note

必須變更下列元件。

- `KeyName`：其中一個 Amazon EC2 金鑰對。
- `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 IDs。
- `Region`：您建立 AD 基礎設施的區域。
- `DomainAddr`：此 IP 地址是 AD 服務的其中一個 DNS 地址。
- `PasswordSecretArn`：包含密碼之秘密的 Amazon Resource Name (ARN) `DomainReadOnlyUser`。
- `BucketName`：存放引導指令碼的儲存貯體名稱。
- `AdditionalPolicies` / `Policy`：讀取網域認證政策 `ReadCertExample` 的 Amazon Resource Name (ARN)。
- `CustomActions` / `OnNodeConfigured` / `Args`：持有網域認證政策之秘密的 Amazon Resource Name (ARN)。

為了提高安全狀態，建議您使用 `HeadNode` / `Ssh` / `AllowedIps` 組態來限制 SSH 對前端節點的存取。

請注意，中指定的憑證 `LdapTlsCaCert` 必須可供所有叢集節點存取。

i 硬性要求

- 中指定的憑證LdapTlsCaCert必須可供所有叢集節點存取。

無法存取憑證的節點將無法從目錄解析使用者。

```
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: amzn-s3-demo-bucket
        EnableWriteAccess: false
        KeyName: bootstrap/active-directory/active-directory.head.post.sh
    CustomActions:
      OnNodeConfigured:
        Script: s3://amzn-s3-demo-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

```

引導指令碼

建立引導檔案之後，在將其上傳至 S3 儲存貯體之前，請執行 `chmod +x active-directory.head.post.sh` 以授予 AWS ParallelCluster 執行許可。

```

#!/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` : 其中一個 Amazon EC2 金鑰對。
- `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路 IDs。
- `Region` : 您建立 AD 基礎設施的區域。

- `DomainAddr` : 此 IP 地址是 AD 服務的其中一個 DNS 地址。
- `PasswordSecretArn` : 包含密碼之秘密的 Amazon Resource Name (ARN)`DomainReadOnlyUser`。

為了提高安全性狀態，建議您使用 `HeadNode/Ssh/AllowedIps` 組態來限制 SSH 對前端節點的存取。

```
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

Note

必須變更下列元件。

- KeyName : 其中一個 Amazon EC2 金鑰對。
- SubnetId / SubnetIds : CloudFormation 快速建立堆疊 (自動教學課程) 或 python 指令碼 (手動教學課程) 輸出中提供的其中一個子網路 IDs。
- Region : 您建立 AD 基礎設施的區域。
- DomainAddr : 此 IP 地址是您 AD 服務的其中一個 DNS 地址。
- PasswordSecretArn : 包含 密碼之秘密的 Amazon Resource Name (ARN)DomainReadOnlyUser。

為了提高安全性狀態，建議您使用 HeadNode/Ssh/AllowedIps 組態來限制 SSH 對前端節點的存取。

```
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": 3.15.0,
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

以使用者身分連線至叢集

您可以使用下列命令來判斷叢集的狀態。

```
$ 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
```

您可以提供在 為新使用者建立的SSH金鑰，不使用密碼登入/home/user000@HEAD_NODE_IP/.ssh/id_rsa。

如果ssh命令成功，您已成功以已驗證使用 Active Directory (AD) 的使用者身分連線至叢集。

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": "3.15.0",
    "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"
```

成功刪除叢集後，請繼續下一個步驟。

自動化

刪除 Active Directory 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 執行個體。
 - a. 在 <https://console.aws.amazon.com/ec2/> 的導覽窗格中，選擇執行個體。
 - b. 從執行個體清單中，選擇您建立以將使用者新增至目錄的執行個體。
 - c. 選擇執行個體狀態，然後選擇終止執行個體。
2. 刪除託管區域。
 - a. recordset-delete.json 使用下列內容建立。在此範例中，HostedZoneId 是負載平衡器的正式託管區域 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 將記錄集變更提交至託管區域。

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNMSJUB \
  --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 接聽程式。

```
$ 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 讀取憑證的政策。

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/  
ReadCertExample
```

7. 刪除包含網域憑證的秘密。

```
$ 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 Active Directory (AD) 資源。

a. 從 python 指令碼 的輸出取得下列資源 IDsad.py :

- AD ID
- AD IDs
- AD VPC ID

b. 執行下列命令來刪除目錄。

```
$ 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. 分離網際網路閘道。

```
$ 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
```



```
--vpc-id vpc-021345abcdef6789 \  
--region region-id
```

- j. 刪除包含ReadOnlyUser密碼的秘密。

```
$ aws secretsmanager delete-secret \  
  --secret-id arn:aws:secretsmanager:region-  
id:123456789012:secret:ADSecretPassword-1234" \  
  --region region-id
```

使用 AWS KMS 金鑰設定共用儲存加密

了解如何設定客戶受管 AWS KMS 金鑰，以加密和保護針對設定的叢集檔案儲存系統中的資料 AWS ParallelCluster。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

AWS ParallelCluster 支援下列共用儲存組態選項：

- [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 for Lustre 共用儲存系統加密提供客戶受管 AWS KMS 金鑰。若要使用這些政策，您必須為下列項目建立和設定 IAM 政策：

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

先決條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有 [Amazon EC2 金鑰對](#)。
- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的[許可](#)。

主題

- [建立政策](#)
- [設定和建立叢集](#)

建立政策

在本教學課程中，您將建立使用 AWS KMS 金鑰設定共用儲存加密的政策。

建立政策

1. 前往 IAM 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政策。請務必 **123456789012** 以您的 AWS 帳戶 ID 和金鑰 Amazon Resource Name (ARN) 取代所有出現的 `123456789012`，並以您自己的 AWS 區域 取代。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:123456789012:key/abcd1234-ef56-gh78-ij90-abcd1234efgh5678"
      ]
    }
  ]
}
```

5. 在本教學課程中，將政策命名為 `ParallelClusterKmsPolicy`，然後選擇建立政策。
6. 記下政策 ARN。您需要它來設定叢集。

設定和建立叢集

以下是範例叢集組態，其中包含具有加密功能的 Amazon Elastic Block Store 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
```

將紅色文字中的項目取代為您自己的值。然後，建立使用 AWS KMS 金鑰在 Amazon EBS 中加密資料的叢集。

Amazon EFS 和 FSx for Lustre 檔案系統的組態類似。

Amazon EFS SharedStorage組態如下所示。

```
...
SharedStorage:
  - MountDir: /shared/efs1
    Name: shared-efs1
    StorageType: Efs
    EfsSettings:
      Encrypted: True
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

FSx for Lustre SharedStorage組態如下所示。

```
...
SharedStorage:
  - MountDir: /shared/fsx1
    Name: shared-fsx1
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      DeploymentType: PERSISTENT_1
      PerUnitStorageThroughput: 200
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

在多個佇列模式叢集中執行任務

本教學課程說明如何 AWS ParallelCluster 在具有[多個佇列模式](#)的 上執行您的第一個「Hello World」任務。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

先決條件

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有 [Amazon EC2 金鑰對](#)。

- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的[許可](#)。

設定您的叢集

首先，執行下列命令，確認 AWS ParallelCluster 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版本 AWS ParallelCluster。

接著，執行 `pcluster configure` 以產生基本組態檔案。遵循此命令的所有提示。

```
$ 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` 會顯示名為 `multi-queue-mode` 的基本組態檔案。此檔案包含基本叢集組態。

在下一個步驟中，您會修改新的組態檔案，並啟動具有多個佇列的叢集。

Note

此教學課程使用的某些執行個體不符合免費方案資格。

在本教學課程中，請修改您的組態檔案以符合下列組態。以紅色反白顯示的項目代表您的組態檔案值。保留您自己的值。

```
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

`multi-queue-cluster` 根據您的組態檔案建立名為 `multi-queue-cluster` 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": "3.15.0",
    "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": "3.15.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

建立叢集時，`clusterStatus` 欄位會顯示 `CREATE_COMPLETE`。

登入前端節點

使用您的私有 SSH 金鑰檔案登入前端節點。

```
$ 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，請執行 `sinfo` 命令來驗證您的排程器佇列是否已設定。

如需的詳細資訊 `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和 `t2.micro` 一個處於可用 `idle` 狀態的 `c5.xlarge` 運算節點。

其他節點都處於省電狀態，以節點狀態的 `~` 尾碼表示，沒有支援它們的 Amazon EC2 執行個體。預設佇列會以佇列名稱後面的尾碼表示。* `spot` 是您的預設任務佇列。

在多個佇列模式中執行任務

接下來，嘗試執行任務以睡眠一段時間。任務稍後會輸出自己的主機名稱。請確定目前使用者可執行此指令碼。

```
$ 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` 命令提交任務。使用 `-N 2` 選項為此任務請求兩個節點，並確認任務已成功提交。如需的詳細資訊 `sbatch`，請參閱 Slurm 文件 [sbatch](#) 中的。

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 1
```

您可以使用 `squeue` 命令檢視佇列並檢查任務的狀態。由於您未指定特定佇列，因此會使用預設佇列 (`spot`)。如需的詳細資訊 `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。等待任務完成。這大約需要 30 秒。然後，`squeue` 再次執行。

```
$ squeue
JOBID PARTITION    NAME      USER  ST        TIME  NODES NODELIST(REASON)
```

現在佇列中的任務都已完成，請尋找 `slurm-1.out` 目前目錄中名為 `slurm-1.out` 的輸出檔案。

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```


輸出顯示任務在 `spot-st-t2micro-1` 和 `spot-st-c5xlarge-1` 節點上成功執行。

現在使用以下命令為特定執行個體指定限制條件，以提交相同的任務。

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 2
```

您針對 使用這些參數 `sbatch`：

- `-N 3`– 請求三個節點。
- `-p spot`– 將任務提交至 `spot` 佇列。您也可以指定 `ondemand`，將任務提交至 `ondemand` 佇列。
- `-C "[c5.xlarge*1&t2.micro*2]"`– 指定此任務的特定節點限制條件。這會請求一個 `c5.xlarge` 節點和兩個 `t2.micro` 節點用於此任務。

執行 `sinfo` 命令以檢視節點和佇列。中的佇列 `AWS ParallelCluster` 在 中稱為分割區 `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]
```

節點正在啟動。這由節點狀態的 # 尾碼表示。執行 `squeue` 命令以檢視叢集中任務的相關資訊。

```
$ 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
```

您的任務處於 `CF`(`CONFIGURING`) 狀態，正在等待執行個體擴展並加入叢集。

大約三分鐘後，節點可用，任務進入 `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
```

任務完成，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，在佇列中沒有任何任務後，請檢查slurm-2.out本機目錄中的。

```
$ 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]
```

登出叢集後，您可以執行 `pcluster delete-cluster` 來清除 pcluster。如需詳細資訊，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"
  }
}
```

使用 AWS ParallelCluster API

在本教學課程中，您會使用 [Amazon API Gateway](#) 和 anCloudFormation 範本建置和測試 API。AWS ParallelCluster CloudFormation 然後，您可以使用 GitHub 上可用的範例用戶端來使用 API。如需如何使用 API 的詳細資訊，請參閱 [AWS ParallelCluster API](#)。

如需詳細資訊，請參閱 EC2 [Image Builder 使用者指南中的使用映像建置器建立自訂元件](#)。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱 [AWS 使用的服務 AWS ParallelCluster](#)。

先決條件

- AWS CLI 會在您的運算環境中 [安裝](#) 和設定。
- 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 \(建議\)](#)。
- 您有 [Amazon EC2 金鑰對](#)。
- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的 [許可](#)。

步驟 1：使用 Amazon API Gateway 建置 API

保留在您的主要使用者目錄中並啟用您的虛擬環境：

1. 安裝實用的 JSON 命令列處理器。

```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. 執行下列命令以取得您的 AWS ParallelCluster 版本，並將其指派給環境變數。

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" |tee -a ~/.bashrc
```

3. 建立環境變數，並將您的區域 ID 指派給該變數。

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" |tee -a ~/.bashrc
```

4. 執行下列命令來部署 API。

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" |tee -a ~/.bashrc
```

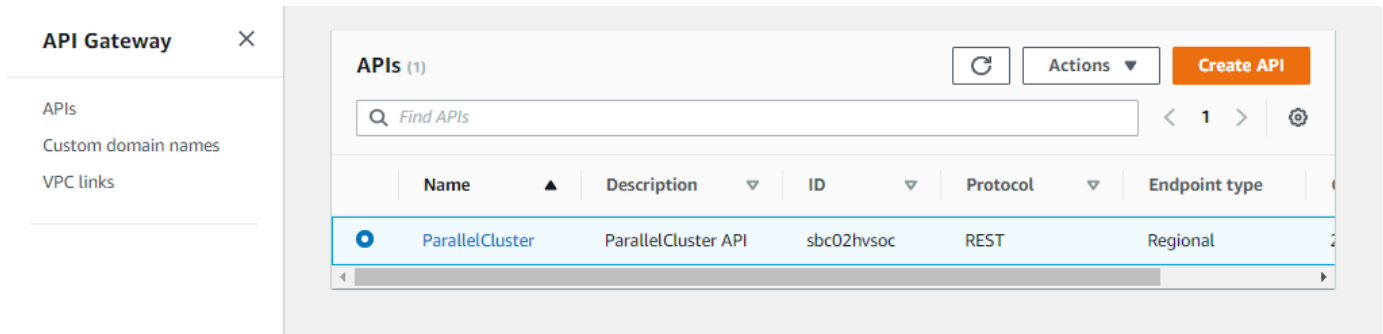
```
aws cloudformation create-stack \
  --region ${AWS_DEFAULT_REGION} \
  --stack-name ${API_STACK_NAME} \
  --template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
parallelcluster-api.yaml \
  --capabilities CAPABILITY_NAMED_IAM 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"
}
```

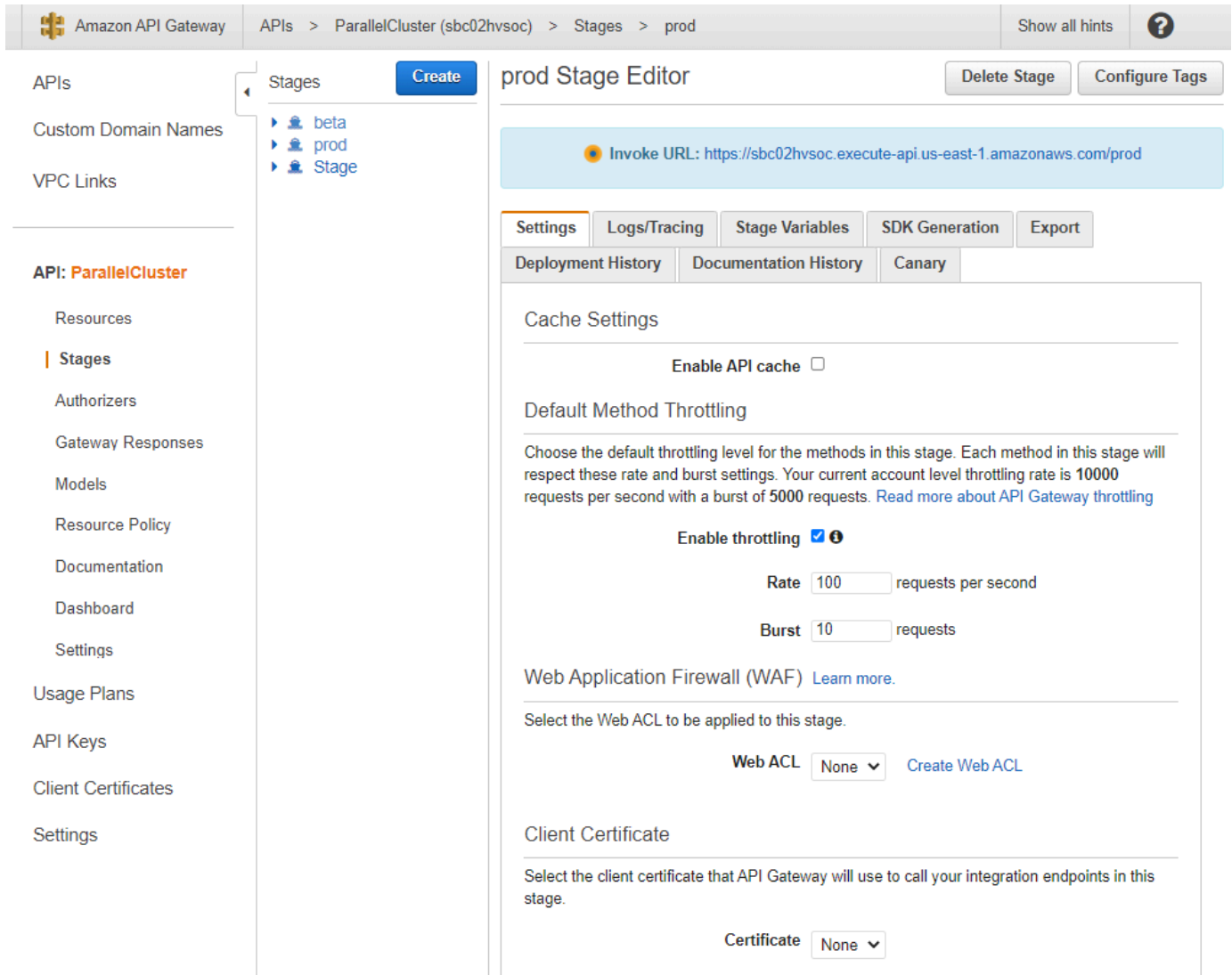
程序完成後，請繼續下一個步驟。

步驟 2：在 Amazon API Gateway 主控台中測試 API

1. 登入 AWS 管理主控台。
2. 導覽至 [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 提供用來存取或叫用 API 的 URL。它以藍色反白顯示。

6. 選擇資源，然後選取 **GET** 下的 **/clusters**。

7. 選擇測試圖示，然後向下捲動並選擇測試圖示。

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.

您的回應 /clusters GET 隨即出現。

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > 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: /v3/clusters
Request: /v3/clusters
Status: 200
Latency: 3203 ms
Response Body

No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.

Query Strings: {clusters} param1=value1¶m2=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 cd api 目錄，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. 導覽回您的主要使用者目錄。

3. 匯出用戶端在執行時所使用的 API Gateway 基本 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. 執行下列命令來存放範例用戶端用來存取 API 的登入資料。

```
$ 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主使用者目錄中test_pcluster_client.py的。用戶端程式碼會發出執行下列動作的請求：

- 建立 叢集
- 描述叢集。
- 列出叢集。
- 描述運算機群。
- 描述叢集執行個體。

```
# 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. 建立叢集組態。

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

3. API 用戶端程式庫會自動偵測來自您環境變數 (例如 `AWS_ACCESS_KEY_ID`、`AWS_SECRET_ACCESS_KEY` 或 `AWS_SESSION_TOKEN`) 或的組態詳細資訊 `$HOME/.aws`。下列命令會將您目前的 IAM 角色切換為指定的 `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"')
```

要監看的錯誤：

如果您看到類似以下的錯誤，表示您已假設 `ParallelClusterApiUserRole` 且您的 `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
```

捨棄角色，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
```

若要為 API 存取提供目前的使用者許可，您必須 [展開資源政策](#)。

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',
 '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請求。

```
# 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. 執行下列命令來刪除叢集。

```

$ 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. 測試完成後，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 管理主控台 或 AWS CLI 來刪除您的 API。

1. 從 CloudFormation 主控台選擇 API 堆疊，然後選擇刪除。
2. 如果使用 `aws`，請執行下列命令 AWS CLI。

使用 CloudFormation。

```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```


使用Slurm會計建立叢集

了解如何使用Slurm會計設定和建立叢集。如需詳細資訊，請參閱[Slurm 使用 會計 AWS ParallelCluster](#)。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的 服務 AWS ParallelCluster](#)。

在本教學課程中，您會使用 [CloudFormation 快速建立範本 \(us-east-1\)](#) 來建立 [Amazon Aurora for MySQL 無伺服器資料庫](#)。範本會指示 CloudFormation 建立所有必要元件，以在與叢集相同的 VPC 上部署無 Amazon Aurora 伺服器資料庫。範本也會為叢集與資料庫之間的連線建立基本聯網和安全性組態。

Note

從 3.3.0 版開始，AWS ParallelCluster 支援使用叢集組態參數 [SlurmSettings](#) / [Database](#) 進行 Slurm 會計。

Note

快速建立範本做為範例。此範本未涵蓋Slurm會計資料庫伺服器的所有可能使用案例。您有責任建立具有適合您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](#)。
- 您有 [Amazon EC2 金鑰對](#)。
- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的[許可](#)。
- 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 [Aurora MySQL 的 Aurora Serverless v2](#)。

步驟 1：建立的 VPC 和子網路 AWS ParallelCluster

若要將提供的 CloudFormation 範本用於 Slurm 會計資料庫，您必須備妥叢集的 VPC。您可以手動或作為[使用 AWS ParallelCluster 命令列界面設定和建立叢集](#)程序的一部分來執行此操作。如果您已使用 AWS ParallelCluster，則可能已準備好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 快速建立範本 \(us-east-1\)](#) 建立用於 Slurm 會計的資料庫堆疊。範本需要以下輸入：

- 資料庫伺服器登入資料，特別是管理員使用者名稱和密碼。
- 無 Amazon Aurora 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登入資料和大小。對於聯網選項，您必須使用 AWS ParallelCluster 部署叢集的相同 VPC。您可以建立資料庫的子網路，並將其做為輸入傳遞至範本。或者，為兩個子網路提供兩個取消連接 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。

建立資料庫伺服器需要幾分鐘的時間。

步驟 3：建立已啟用 Slurm 會計的叢集

提供的 CloudFormation 範本會產生具有一些已定義輸出的 CloudFormation 堆疊。從 AWS 管理主控台，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 會計，AWS ParallelCluster 叢集組態檔案中必須使用其中一些輸出：

- DatabaseHost：用於 [SlurmSettings / Database / Uri](#) 叢集組態參數。
- DatabaseAdminUser：用於 [SlurmSettings // Database UserName](#) 叢集組態參數值。
- DatabaseSecretArn：用於 [SlurmSettings / Database / PasswordSecretArn](#) 叢集組態參數。
- DatabaseClientSecurityGroup：這是連接到叢集前端節點的安全群組，其定義在 [Networking / HeadNode SecurityGroups](#) 組態參數中。

使用輸出值更新您的叢集組態檔案 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會計命令，例如 `sacctmgr` 或 `sacct`。

使用外部Slurmdbd會計建立叢集

了解如何設定和建立具有外部Slurmdbd會計的叢集。如需詳細資訊，請參閱[Slurm使用 進行會計 AWS ParallelCluster](#)。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為 AWS 建立或更新 AWS ParallelCluster 映像和叢集時建立的資源付費。如需詳細資訊，請參閱 [AWS 所使用的服務 AWS ParallelCluster](#)。

UI AWS ParallelCluster 是以無伺服器架構為基礎，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，以在與叢集相同的 VPC 上部署 Slurmdbd 執行個體。範本會為叢集與資料庫之間的連線建立基本聯網和安全性組態。

Note

從開始 version 3.10.0，使用叢集組態參數 `AWS ParallelCluster` 支援外部 `SlurmdbdSlurmSettings / ExternalSlurmdbd`。

Note

快速建立範本做為範例。此範本未涵蓋所有可能的使用案例。您有責任使用適合您生產工作負載的組態和容量建立外部 Slurmdbd。

事前準備：

- AWS ParallelCluster [已安裝](#)。
- AWS CLI [已安裝並設定](#)。
- 您有 [Amazon Elastic Compute Cloud 金鑰對](#)。
- 您的 AWS Identity and Access Management 角色具有執行 CLI `pcluster` 所需的[許可](#)。

- 您有Slurm會計資料庫。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。

步驟 1：建立 Slurmdbd 堆疊

在本教學課程中，使用 [CloudFormation 快速建立範本 \(us-east-1\)](#) 來建立 Slurmdbd 堆疊。範本需要以下輸入：

聯網

- VPCId：啟動 Slurmdbd 執行個體的 VPC ID。
- SubnetId：啟動 Slurmdbd 執行個體的子網路 ID。
- PrivatePrefix：VPC 的 CIDR 字首。
- PrivateIp：指派給 Slurmdbd 執行個體的次要私有 IP。

資料庫連線

- DBMSClientSG：要連接到 Slurmdbd 執行個體的安全群組。此安全群組應允許資料庫伺服器與 Slurmdbd 執行個體之間的連線。
- DBMSDatabaseName：資料庫的名稱。
- DBMSUsername：資料庫的使用者名稱。
- DBMSPasswordSecretArn：包含資料庫密碼的秘密。
- DBMSUri：資料庫伺服器的 URI。

執行個體設定

- InstanceType：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 金鑰對。

Slurmdbd 設定

- AMIID：Slurmdbd 執行個體的 AMI。AMI 應為 ParallelCluster AMI。ParallelCluster AMI 的版本會決定 Slurmdbd 的版本。
- MungeKeySecretArn：包含 munge 金鑰的秘密，用於驗證 Slurmdbd 和叢集之間的通訊。
- SlurmdbdPort：slurmdbd 使用的連接埠號碼。

- `EnableSlurmdbSystemService`：啟用 `slurmdbd` 做為系統服務，並在執行個體啟動時執行。

Warning

如果資料庫是由不同版本的所建立SlurmDB，請勿使用 `Slurmdbd`做為系統服務。
如果資料庫包含大量項目，`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不會回應。
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已啟用外部的叢集

提供的 CloudFormation 範本會產生具有一些已定義輸出的 CloudFormation 堆疊。

從 AWS 管理主控台檢視 CloudFormation 堆疊中的輸出索引標籤，以檢閱建立的實體。若要啟用 Slurm會計，必須在 AWS ParallelCluster 組態檔案中使用其中一些輸出：

- `SlurmdbdPrivatelp`：用於 [SlurmSettings/ExternalSlurmdbd/主機叢集組態](#) 參數。
- `SlurmdbdPort`：用於 [SlurmSettings/ExternalSlurmdbd/連接埠](#) 叢集組態參數值。
- `AccountingClientSecurityGroup`：這是連接到叢集前端節點的安全群組，其定義在 [HeadNode / Networking / AdditionalSecurityGroups](#) 組態參數中。

此外，從 CloudFormation 堆疊檢視中的參數索引標籤：

- `MungeKeySecretArn`：用於 [SlurmSettings / MungeKeySecretArn](#) 叢集組態參數值。

使用輸出值更新您的叢集組態檔案資料庫參數。使用叢集 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會計命令，例如 `sacctmgr`或 `sacct`。

Warning

`ParallelCluster` 與外部 之間的流量SlurmDB不會加密。建議在信任的網路SlurmDB中執行叢集和外部。

還原至先前的 AWS Systems Manager 文件版本

了解如何還原至先前的 AWS Systems Manager 文件版本。如需 SSM 文件的詳細資訊，請參閱 [AWS Systems Manager 使用者指南](#) 中的 Systems Manager 文件。

使用 AWS ParallelCluster 命令列界面 (CLI) 或 API 時，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱 [AWS 使用的服務 AWS ParallelCluster](#)。

事前準備：

- AWS 帳戶 具有管理 SSM 文件許可的。
- AWS CLI [已安裝並設定](#)。

還原至先前的 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 文件還原至先前的版本。在此範例中，我們會還原至 SessionManagerRunShell 文件的先前版本。您可以使用 SSM SessionManagerRunShell 文件來自訂您啟動的每個 SSM shell 工作階段。

- a. SessionManagerRunShell 執行下列命令來尋找的 DocumentVersion 參數：

```
$ 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. 執行下列命令以還原至先前的版本：

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-  
version 2
```

3. 再次執行 describe-document 命令，確認文件版本已還原：

```
$ 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。

使用 建立叢集 CloudFormation

了解如何使用 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 映像和叢集時建立 AWS 的資源付費。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](#)。
- [Amazon EC2 金鑰對](#)。
- 具有執行 CLI 所需[許可](#)的 IAM [pcluster](#) 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 CloudFormation 範本，以建立叢集和下列 AWS 資源：

- 使用 CloudFormation 快速建立堆疊建立的根 CloudFormation 堆疊。
- 巢狀 CloudFormation 堆疊，其中包含預設政策、預設 VPC 設定和自訂資源提供者。
- 您可以登入並執行任務的範例 AWS ParallelCluster 叢集堆疊和叢集。

使用 建立叢集 AWS CloudFormation

1. 登入 AWS 管理主控台。
2. 開啟 CloudFormation [快速建立連結](#)，以在 CloudFormation 主控台中建立下列資源：
 - 具有公有和私有子網路的 VPC 的巢狀 CloudFormation 堆疊，分別用於執行叢集前端節點和運算節點。
 - 巢狀 CloudFormation 堆疊，具有用於管理叢集的 AWS ParallelCluster 自訂資源。
 - 巢狀 CloudFormation 堆疊，具有管理叢集的預設政策。
 - 巢狀堆疊的根 CloudFormation 堆疊。
 - 具有Slurm排程器和已定義運算節點數量的 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. 針對 KeyName，輸入 Amazon EC2 金鑰對的名稱。
 - b. 針對 AvailabilityZone，為您的叢集節點選擇 AZ，例如 us-east-1a。

4. 勾選方塊以確認頁面底部的每個存取功能。
5. 選擇建立堆疊。
6. 等待 CloudFormation 堆疊達到 CREATE_COMPLETE 狀態。

使用 CloudFormation 命令列界面 (CLI) 建立叢集

在本教學課程中，您會使用 CloudFormation 的 AWS 命令列界面 (CLI) 來部署建立叢集的 CloudFormation 範本。

建立下列 AWS 資源：

- 使用 CloudFormation 快速建立堆疊建立的根 CloudFormation 堆疊。
- 巢狀 CloudFormation 堆疊，其中包含預設政策、預設 VPC 設定和自訂資源提供者。
- 您可以登入並執行任務的範例 AWS ParallelCluster 叢集堆疊和叢集。

以您自己的值取代以#####，例如###。

使用 建立叢集 AWS CloudFormation

1. 使用下列內容建立名為 cluster_template.yaml 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: 3.15.0

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 命令來部署 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 屬性可讓您從叢集建立和更新操作存取驗證訊息。

- 導覽至 [CloudFormation 主控台](#)，然後選取包含 AWS ParallelCluster 自訂資源的堆疊。
- 選擇堆疊詳細資訊，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。如需如何擷取日誌的詳細資訊，請參閱 [AWS ParallelCluster 故障診斷](#)。

存取您的叢集

存取叢集。

ssh 到叢集前端節點

- CloudFormation 堆疊部署完成後，請使用下列命令取得前端節點的 IP 地址：

```
$ 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 主控台叢集堆疊輸出索引標籤中的 HeadNodeIp 參數擷取前端節點 IP 地址。

您可以在此處找到前端節點 IP 地址，因為它已新增至叢集 CloudFormation 範本的 Outputs 區段，特別是此範例叢集。

2. 執行下列命令以連接至叢集前端節點：

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

清除

刪除叢集。

1. 執行下列 AWS CLI 命令來刪除 CloudFormation 堆疊和叢集。

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. 執行下列命令來檢查堆疊刪除狀態。

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

使用 Terraform 部署 ParallelCluster API

在本教學課程中，您將定義簡單的 Terraform 專案來部署 ParallelCluster API。

先決條件

- 已安裝 Terraform v1.5.7+。
- 具有部署 ParallelCluster API 許可的 IAM 角色。請參閱 [the section called “所需的許可”](#)。

定義 Terraform 專案

在本教學課程中，您將定義 Terraform 專案。

1. 建立名為的目錄my-pcluster-api。

您建立的所有檔案都將在此目錄中。

2. 建立檔案provider.tf以設定 AWS 提供者。

```
provider "aws" {  
  region = var.region  
  profile = var.profile  
}
```

3. 建立檔案main.tf以使用 ParallelCluster 模組定義資源。

```
module "parallelcluster_pcluster_api" {  
  source = "aws-tf/parallelcluster/aws//modules/pcluster_api"  
  version = "1.1.0"  
  
  region = var.region  
  api_stack_name = var.api_stack_name  
  api_version = var.api_version  
  
  parameters = {  
    EnableIamAdminAccess = "true"  
  }  
}
```

4. 建立檔案variables.tf以定義可為此專案注入的變數。

```
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。

以下檔案 `us-east-1` 會使用堆疊名稱 在 中部署 ParallelCluster API

3.11.1MyParallelClusterAPI-3111。您將可以使用其堆疊名稱來參考此 ParallelCluster API 部署。

Note

下列程式碼中的 `api_version` 指派可以取代為任何支援的 AWS ParallelCluster 版本。

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-3111"
api_version = "3.11.1"
```

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
}
```

專案目錄為：

```
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 Terraform 命令。

1. 建置專案：

```
terraform init
```

2. 定義部署計畫：

```
terraform plan -out tfplan
```

3. 部署計畫：

```
terraform apply tfplan
```

所需的許可

您需要下列許可，才能使用 Terraform 部署 ParallelCluster API：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:GetTemplate"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormationRead"
    },
    {
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:CreateChangeSet"
      ],
```

```

        "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/
MyParallelClusterAPI*",
        "Effect": "Allow",
        "Sid": "CloudFormationWrite"
    },
    {
        "Action": [
            "cloudformation:CreateChangeSet"
        ],
        "Resource": [
            "arn:aws:cloudformation:us-east-1:111122223333:aws:transform/
Include",
            "arn:aws:cloudformation:us-east-1:111122223333:aws:transform/
Serverless-2016-10-31"
        ],
        "Effect": "Allow",
        "Sid": "CloudFormationTransformWrite"
    },
    {
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:us-east-1:111122223333:*-aws-parallelcluster/
parallelcluster/*/api/ParallelCluster.openapi.yaml",
            "arn:aws:s3:us-east-1:111122223333:*-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:aws:iam:111122223333:role/*",
        "arn:aws:iam:111122223333:policy/*"
    ],
    "Effect": "Allow",
    "Sid": "IAM"
},
{
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam:111122223333:role/ParallelClusterLambdaRole-*",
        "arn:aws:iam:111122223333: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:aws:lambda:us-east-1:111122223333:layer:PCLayer-*",
        "arn:aws:lambda:us-east-1:111122223333:function:*-
ParallelClusterFunction-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "logs:CreateLogGroup",
        "logs>DeleteLogGroup",
        "logs:DescribeLogGroups",

```

```

        "logs:PutRetentionPolicy",
        "logs:TagLogGroup",
        "logs:UntagLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:us-east-1:111122223333:log-group:/aws/lambda/*-
ParallelClusterFunction-*"
    ],
    "Effect": "Allow",
    "Sid": "Logs"
},
{
    "Action": [
        "apigateway:DELETE",
        "apigateway:GET",
        "apigateway:PATCH",
        "apigateway:POST",
        "apigateway:PUT",
        "apigateway:UpdateRestApiPolicy"
    ],
    "Resource": [
        "arn:aws:apigateway:us-east-1::/restapis",
        "arn:aws:apigateway:us-east-1::/restapis/*",
        "arn:aws:apigateway:us-east-1::/tags/*"
    ],
    "Effect": "Allow",
    "Sid": "APIGateway"
}
]
}

```

使用 Terraform 建立叢集

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。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 Terraform 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 Terraform 部署 ParallelCluster API”](#)。

- 具有叫用 ParallelCluster API 許可的 IAM 角色。請參閱 **【必要許可】**

定義 Terraform 專案

在本教學課程中，您將定義簡單的 Terraform 專案來部署叢集。

1. 建立名為的目錄my-clusters。

您建立的所有檔案都將在此目錄中。

2. 建立檔案terraform.tf以匯入 ParallelCluster 供應商。

```
terraform {
  required_version = ">= 1.5.7"
  required_providers {
    aws-parallelcluster = {
      source = "aws-tf/aws-parallelcluster"
      version = "~> 1.0"
    }
  }
}
```

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. 建立檔案main.tf以使用 ParallelCluster 模組定義資源。

```
module "pcluster" {
  source = "aws-tf/parallelcluster/aws"
  version = "1.1.0"

  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，將多個叢集定義為 Terraform 本機變數。

Note

您可以在 `cluster_config` 元素中定義多個叢集。對於每個叢集，您可以明確定義本機變數內的叢集屬性（請參閱 `DemoCluster01`）或參考外部檔案（請參閱 `DemoCluster02`）。

若要檢閱您可以在組態元素中設定的叢集屬性，請參閱 [the section called “叢集組態檔案”](#)。

若要檢閱您可以為建立叢集設定的選項，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，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. 建立檔案 `config/cluster_config.yaml`，這是可插入 Terraform 變數的標準 ParallelCluster 組態檔案。

若要檢閱您可以在組態元素中設定的叢集屬性，請參閱 [the section called “叢集組態檔案”](#)。

```

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` 以定義可注入叢集組態的變數。

此檔案可讓您定義可在叢集組態中使用的動態值，例如區域和子網路。

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它們。

```
locals {
  config_vars = {
    subnet = var.subnet_id
    region = var.cluster_region
  }
}
```

9. 建立檔案variables.tf以定義可為此專案注入的變數。

```
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。

以下檔案 subnet-123456789 使用已部署在 us-east-1 堆疊名稱為 的現有 ParallelCluster API 3.11.1，在 eu-west-1 子網路 中部署叢集。 MyParallelClusterAPI-3111

```
region = "us-east-1"  
api_stack_name = "MyParallelClusterAPI-3111"  
api_version = "3.11.1"  
  
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  
}
```

專案目錄為：

```
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 Terraform 命令。

Note

此範例假設您已在帳戶中部署 ParallelCluster API。

1. 建置專案：

```
terraform init
```

2. 定義部署計畫：

```
terraform plan -out tfplan
```

3. 部署計畫：

```
terraform apply tfplan
```

使用叢集部署 ParallelCluster API

如果您尚未部署 ParallelCluster API，而且想要使用叢集部署它，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
  parameters = {
    EnableIamAdminAccess = "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

您需要下列許可，才能使用 Terraform 部署叢集：

- 擔任 ParallelCluster API 角色，負責與 ParallelCluster API 互動
- 描述 ParallelCluster API 的 CloudFormation 堆疊，以驗證其存在並擷取其參數和輸出

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:sts::111122223333:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}
```

```
}
```

使用 Terraform 建立自訂 AMI

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。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 Terraform 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 Terraform 建立叢集”](#)。
- 具有叫用 ParallelCluster API 許可的 IAM 角色。請參閱 [the section called “所需的許可”](#)。

定義 Terraform 專案

在本教學課程中，您將定義簡單的 Terraform 專案來部署 ParallelCluster 自訂 AMI。

1. 建立名為的目錄my-amis。

您建立的所有檔案都將在此目錄中。

2. 建立檔案terraform.tf以匯入 ParallelCluster 供應商。

```
terraform {
  required_version = ">= 1.5.7"
  required_providers {
    aws-parallelcluster = {
      source = "aws-tf/aws-parallelcluster"
      version = "~> 1.0"
    }
  }
}
```

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. 建立檔案main.tf以使用 ParallelCluster 模組定義資源。

若要檢閱您可以在 image_configuration 元素中設定的映像屬性，請參閱 [the section called “建置映像組態檔案”](#)。

若要檢閱您可以為影像建立設定的選項，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. 建立檔案variables.tf以定義可為此專案注入的變數。

```
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. 建立檔案 `terraform.tfvars` 以設定變數的任意值。

使用下列檔案，使用已部署在 `us-east-1` 堆疊名為 `MyParallelClusterAPI-3111` 的現有 ParallelCluster API 3.11.1，`us-east-1` 以適用於 `x86_64` 架構的 Amazon Linux 2 在中部署自訂 AMI。

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-3111"
api_version = "3.11.1"

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
}
```

專案目錄為：

```
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 Terraform 命令。

1. 建置專案：

```
terraform init
```

2. 定義部署計畫：

```
terraform plan -out tfplan
```

3. 部署計畫：

```
terraform apply tfplan
```

所需的許可

您需要下列許可，才能使用 Terraform 部署自訂 AMI：

- 擔任 ParallelCluster API 角色，負責與 ParallelCluster API 互動
- 描述 ParallelCluster API 的 CloudFormation 堆疊，以驗證其是否存在並擷取其參數和輸出

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:sts::111122223333:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}
```

AWS ParallelCluster UI 與 Identity Center 整合

本教學課程的目標是示範如何針對單一登入解決方案整合 AWS ParallelCluster UI 與 IAM Identity Center，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 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

事前準備：

- 可依照此處的指示安裝的現有 AWS ParallelCluster UI。 [???](#)
- 現有的受管 Active Directory，最好也用於[整合 AWS ParallelCluster](#)。

啟用 IAM Identity Center

如果您已將身分中心連接到您的 AWS Managed Microsoft AD (Active Directory)，則可以使用它，也可以跳到將應用程式新增至 IAM Identity Center 一節。

如果您還沒有連接到 的身分中心 AWS Managed Microsoft AD，請依照下列步驟進行設定。

啟用身分中心

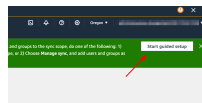
1. 在主控台中，導覽至 IAM Identity Center。（請確定您位於您擁有的區域 AWS Managed Microsoft AD。）
2. 按一下啟用按鈕，這可能會詢問您是否要啟用組織，這是必要項目，因此您可以選取 來啟用它。注意：這將傳送電子郵件給您帳戶的管理員，其中包含您應遵循連結確認的確認電子郵件。

將 Identity Center 連線至 Managed AD

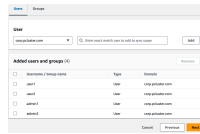
1. 在啟用身分中心後的下一頁，您應該會看到建議的設定步驟，在步驟 1 下，選取選擇您的身分來源。
2. 在身分來源區段中，按一下動作下拉式選單（右上角），然後選取變更身分來源。
3. 選取 Active Directory。
4. 在現有目錄下，選擇您的目錄。
5. 按一下 Next (下一步)。
6. 檢閱您的變更，捲動至底部，在文字方塊中輸入 ACCEPT 進行確認，然後按一下變更身分來源。
7. 等待變更完成，然後您應該會在頂端看到綠色橫幅。

將使用者和群組同步至 Identity Center

1. 在綠色橫幅中，按一下開始引導設定（右上角的按鈕）



2. 在設定屬性映射中，按一下下一步
3. 在設定同步範圍區段中，輸入您要同步至身分中心的使用者名稱，然後按一下新增
4. 新增使用者和群組完成後，請按一下下一步



5. 檢閱您的變更，然後按一下儲存組態
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 > 啟用使用者存取

注意：如果您的頂端有警告橫幅，然後等待使用者同步（嘗試重新整理按鈕以查看是否已同步），您可能需要選取繼續同步。



將應用程式新增至 IAM Identity Center

將使用者與 IAM Identity Center 同步後，您將需要新增應用程式。這會設定哪些啟用 SSO 的應用程式可從 IAM Identity Center 入口網站使用。在此情況下，我們將新增 AWS ParallelCluster UI 做為應用程式，而 IAM Identity Center 將做為身分提供者。

下一個步驟會將 AWS ParallelCluster UI 新增為 IAM Identity Center 中的應用程式。AWS ParallelCluster UI 是一種 Web 入口網站，可協助使用者管理其叢集。如需詳細資訊，請參閱 [AWS ParallelCluster UI](#)。

在 Identity Center 中設定應用程式

1. 在 IAM Identity Center > 應用程式下（位於左側選單列，按一下應用程式）
2. 按一下新增應用程式
3. 選取新增自訂 SAML 2.0 應用程式
4. 按一下下一步
5. 選取您想要使用的顯示名稱和描述（例如 PCUI 和 AWS ParallelCluster UI）
6. 在 IAM Identity Center 中繼資料下，複製 IAM Identity Center SAML 中繼資料檔案的連結並儲存以供稍後使用，這將在 Web 應用程式中設定 SSO 時使用
7. 在應用程式屬性下，在應用程式啟動 URL 中，放置您的 PCUI 地址。您可以前往 CloudFormation 主控台、選取對應至 PCUI 的堆疊（例如 parallelcluster-ui），以及前往輸出索引標籤尋找 ParallelClusterUIUrl

例如 <https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com>

8. 在應用程式中繼資料下，選擇手動輸入中繼資料值。然後提供下列值。
 - a. 重要：請務必將 domain-prefix、region 和 userpool-id 值取代為您環境的特定資訊。
 - b. 開啟 Amazon Cognito > 使用者集區主控台，即可取得網域字首、區域和使用者集區 ID



- c. 選取對應至 PCUI 的使用者集區（其將具有使用者集區名稱，例如 pcui-cd8a2-Cognito-153EK3TO45S98-userpool）
 - d. 導覽至應用程式整合



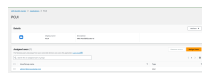
9. 應用程式聲明消費者服務 (ACS) URL：`https://https://<domain-prefix>.auth.<region>.amazoncognito.com/saml2/idpresponse`

應用程式 SAML 對象：`urn:amazon:cognito:sp:<userpool-id>`

10. 選擇提交。然後，前往您新增之應用程式的詳細資訊頁面。
11. 選取動作下拉式清單，然後選擇編輯屬性映射。然後，提供下列屬性。
 - a. 應用程式的使用者屬性：主體（注意：主體已預先填入。）→ 映射至 IAM Identity Center 中的此字串值或使用者屬性：`${user:email}`，格式：`emailAddress`
 - b. 應用程式的使用者屬性：電子郵件 → 映射到此字串值或 IAM Identity Center 中的使用者屬性：`${user:email}`，格式：未指定



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 Active Directory 中可存取 PCUI 介面的使用者。



在使用者集區中將 IAM Identity Center 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



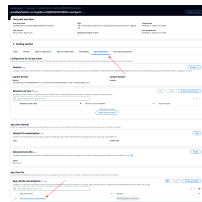
2. 選擇 SAML IdP
3. 對於提供者名稱，請提供 IdentityCenter
4. 在中繼資料文件來源下，選擇輸入中繼資料文件端點 URL，並提供在 Identity Center 的應用程式設定期間複製的 URL
5. 在屬性下，針對電子郵件選擇電子郵件



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

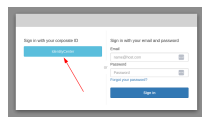
1. 接著，在使用者集區的應用程式整合區段下，選擇應用程式用戶端清單下列出的用戶端



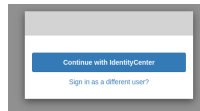
2. 在託管 UI 下選擇編輯
3. 在身分提供者下，也選擇 IdentityCenter。
4. 選擇 Save changes (儲存變更)

驗證您的設定

1. 接下來，我們將透過登入 PCUI 驗證剛建立的設定。登入 PCUI 入口網站，現在您應該會看到使用公司 ID 登入的選項：

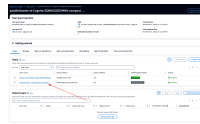


2. 按一下 IdentityCenter 按鈕應該會帶您前往 IAM Identity Center IdP 登入，後面接著包含 PCUI 的應用程式頁面，開啟該應用程式。
3. 進入下列畫面後，您的使用者將新增至 Cognito 使用者集區。



讓您的使用者成為管理員

1. 現在導覽至 Amazon Cognito > 使用者集區主控台，然後選取應具有 identitycenter 字首的新建立使用者



2. 在群組成員資格下，選取新增使用者至群組，選擇管理員並按一下新增。
3. 現在當您按一下繼續 IdentityCenter 時，將會導覽至 UI AWS ParallelCluster 頁面。

使用 Pyxis 執行容器化任務

了解如何建立叢集，該叢集可以使用「Pyxis」執行容器化任務，這是一種 SPANK 外掛程式，可在 SLURM 中管理容器化任務。Pyxis 中的容器由 Enroot 管理，這是將傳統容器/OS 映像轉換為無權限沙盒的工具。如需詳細資訊，請參閱 [NVIDIA Pyxis](#) 和 [NVIDIA Enroot](#)。

Note

- 此功能適用於 AWS ParallelCluster v3.11.1
- 本教學課程中的指令碼會移動 (mv) 一些檔案，從原始位置將其刪除。如果您想要將這些檔案的副本保留在原始位置，請變更指令碼以改用 copy (cp) 命令。

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。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](#)。
- [Amazon EC2 金鑰對](#)。
- 具有執行叢集 CLI 所需許可的 IAM 角色。 <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>

建立叢集

從 AWS ParallelCluster 3.11.1 開始，所有官方 AMIs 都隨附預先安裝的 Pyxis 和 Enroot。特別是，SLURM 會重新編譯為使用「Pyxis 支援」，Enroot 會安裝為系統中的二進位檔。不過，您必須根據您的特定需求進行設定。Enroot 和 Pyxis 使用的資料夾將對叢集效能產生重大影響。如需詳細資訊，請參閱 [Pyxis 文件](#) 和 [Enroot 文件](#)。

為方便起見，您可以在 `parallelcluster/examples/` 中找到適用於 Pyxis、Enroot 和 SPANK 的範例組態。

若要使用我們提供的範例組態來部署叢集，請完成下列教學課程。

使用範例組態建立叢集

必須先為 Enroot 建立持久性和揮發性目錄，然後為 Pyxis 建立執行期目錄，最後在整個叢集中啟用 Pyxis 做為 SPANK 外掛程式，才能在前端節點上設定 Pyxis 和 Enroot。

1. 在前端節點中執行以下指令碼做為 [OnNodeConfigured](#) 自訂動作，以在前端節點上設定 Pyxis 和 Enroot。

```
#!/bin/bash
set -e

echo "Executing $0"

# Configure Enroot
ENROOT_PERSISTENT_DIR="/var/enroot"
ENROOT_VOLATILE_DIR="/run/enroot"

sudo mkdir -p $ENROOT_PERSISTENT_DIR
sudo chmod 1777 $ENROOT_PERSISTENT_DIR
sudo mkdir -p $ENROOT_VOLATILE_DIR
sudo chmod 1777 $ENROOT_VOLATILE_DIR
sudo mv /opt/parallelcluster/examples/enroot/enroot.conf /etc/enroot/enroot.conf
sudo chmod 0644 /etc/enroot/enroot.conf

# Configure Pyxis
PYXIS_RUNTIME_DIR="/run/pyxis"

sudo mkdir -p $PYXIS_RUNTIME_DIR
sudo chmod 1777 $PYXIS_RUNTIME_DIR
```

```
sudo mkdir -p /opt/slurm/etc/pluginstack.conf.d/
sudo mv /opt/parallelcluster/examples/spank/pluginstack.conf /opt/slurm/etc/
sudo mv /opt/parallelcluster/examples/pyxis/pyxis.conf /opt/slurm/etc/
pluginstack.conf.d/
sudo -i scontrol reconfigure
```

2. 在運算機群上，必須透過為 Enroot 建立持久性和揮發性目錄，以及為 Pyxis 建立執行時間目錄，來設定 Pyxis 和 Enroot。在運算節點中執行以下指令碼做為 [OnNodeStart](#) 自訂動作，以在運算機群上設定 Pyxis 和 Enroot。

```
#!/bin/bash
set -e

echo "Executing $0"

# Configure Enroot
ENROOT_PERSISTENT_DIR="/var/enroot"
ENROOT_VOLATILE_DIR="/run/enroot"
ENROOT_CONF_DIR="/etc/enroot"

sudo mkdir -p $ENROOT_PERSISTENT_DIR
sudo chmod 1777 $ENROOT_PERSISTENT_DIR
sudo mkdir -p $ENROOT_VOLATILE_DIR
sudo chmod 1777 $ENROOT_VOLATILE_DIR
sudo mkdir -p $ENROOT_CONF_DIR
sudo chmod 1777 $ENROOT_CONF_DIR
sudo mv /opt/parallelcluster/examples/enroot/enroot.conf /etc/enroot/enroot.conf
sudo chmod 0644 /etc/enroot/enroot.conf

# Configure Pyxis
PYXIS_RUNTIME_DIR="/run/pyxis"

sudo mkdir -p $PYXIS_RUNTIME_DIR
sudo chmod 1777 $PYXIS_RUNTIME_DIR

# In Ubuntu24.04 Apparmor blocks the creation of unprivileged user namespaces,
# which is required by Enroot. So to run Enroot, it is required to disable this
# restriction.
# See https://ubuntu.com/blog/ubuntu-23-10-restricted-unprivileged-user-namespaces
source /etc/os-release
```



```
if [ "${ID}${VERSION_ID}" == "ubuntu24.04" ]; then
    echo "kernel.apparmor_restrict_unprivileged_userns = 0" | sudo tee /etc/
sysctl.d/99-pcluster-disable-apparmor-restrict-unprivileged-userns.conf
    sudo sysctl --system
fi
```

提交任務

現在，Pyxis 已在您的叢集中設定，您可以使用 `sbatch` and `srun` 命令來提交容器化任務，這些任務現在已擴充容器特定選項。

```
# Submitting an interactive job
srun -N 2 --container-image docker://ubuntu:22.04 hostname

# Submitting a batch job
sbatch -N 2 --wrap='srun --container-image docker://ubuntu:22.04 hostname'
```

使用啟用 EFA 的 FSx Lustre 建立叢集

在本教學課程中，您將建立使用啟用 EFA 的 FSx Lustre 檔案系統做為共用儲存體的叢集。在啟用 EFA 的情況下使用 FSx Lustre 檔案系統，可將效能提升高達 8 倍。若要驗證已啟用 EFA 的檔案系統是否為您需要的檔案系統，請參閱《FSx for Lustre 使用者指南》中的[使用已啟用 EFA 的檔案系統](#)。

使用時 AWS ParallelCluster，您只需為建立或更新 AWS ParallelCluster 映像和叢集時建立 AWS 的資源付費。如需詳細資訊，請參閱[AWS 使用的服務 AWS ParallelCluster](#)。

需求

- [已安裝並設定](#) AWS CLI。
- [已安裝並設定](#) ParallelCluster CLI。
- 用於登入叢集的 [Amazon EC2 金鑰對](#)。
- 具有執行 ParallelCluster CLI 所需[許可](#)的 IAM 角色。

建立安全群組

在將部署叢集和檔案系統的相同 VPC 中建立兩個安全群組：一個用於叢集節點上執行的用戶端，另一個用於檔案系統。

```
# Create security group for the FSx client
aws ec2 create-security-group \
  --group-name Fsx-Client-SecurityGroup \
  --description "Allow traffic for the FSx Lustre client" \
  --vpc-id vpc-cluster \
  --region region

# Create security group for the FSx file system
aws ec2 create-security-group \
  --group-name Fsx-FileSystem-SecurityGroup \
  --description "Allow traffic for the FSx Lustre File System" \
  --vpc-id vpc-cluster \
  --region region
```

在教學課程的其餘部分中，我們將假設 `sg-client` 和 `sg-file-system` 分別是用戶端和檔案系統的安全群組 ID。

根據 [EFA 的要求](#)，設定用戶端的安全群組，以允許所有傳出流量流向檔案系統。

```
# Allow all outbound traffic from the client to the file system
aws ec2 authorize-security-group-egress \
  --group-id sg-client \
  --protocol -1 \
  --port -1 \
  --source-group sg-file-system \
  --region region
```

設定檔案系統的安全群組，以允許其本身的所有傳入/傳出流量，以及 [EFA 要求](#) 的所有來自用戶端的傳入流量。

```
# Allow all inbound traffic within this security group
aws ec2 authorize-security-group-ingress \
  --group-id sg-file-system \
  --protocol -1 \
  --port -1 \
  --source-group sg-file-system \
```

```
--region region

# Allow all outbound traffic within this security group
aws ec2 authorize-security-group-egress \
  --group-id sg-file-system \
  --protocol -1 \
  --port -1 \
  --source-group sg-file-system \
  --region region

# Allow all inbound traffic from the client
aws ec2 authorize-security-group-ingress \
  --group-id sg-file-system \
  --protocol -1 \
  --port -1 \
  --source-group sg-client \
  --region region

# Allow all outbound traffic to the client
aws ec2 authorize-security-group-egress \
  --group-id sg-file-system \
  --protocol -1 \
  --port -1 \
  --source-group sg-client \
  --region region
```

建立檔案系統

在運算節點所在的相同可用區域 (AZ) 內建立檔案系統；並在下列程式碼中 *subnet-compute-nodes* 以其 ID 取代。這是允許 EFA 與您的檔案系統搭配使用的必要條件。請注意，在建立檔案系統的過程中，我們使用 `EfaEnabled` 屬性啟用 EFA。

```
aws fsx create-file-system \
  --file-system-type LUSTRE \
  --storage-capacity 38400 \
  --storage-type SSD \
  --subnet-ids subnet-compute-nodes \
  --security-group-ids sg-file-system \
  --lustre-configuration
DeploymentType=PERSISTENT_2,PerUnitStorageThroughput=125,EfaEnabled=true,MetadataConfiguration
\
  --region region
```

請記下上一個命令傳回的檔案系統 ID。在教學課程的其餘部分中，*fs-id* 使用此檔案系統 ID 取代。

建立叢集

1. 使用 YAML 組態檔案中設定的 AWS ParallelCluster 下列組態建立叢集：

- a. 以支援的作業系統為基礎的 AMI，例如 Ubuntu 22.04。
- b. 運算節點必須使用具有 [Nitro v4+](#) 的 [EFA 支援的執行個體類型](#)，例如 g6.16xlarge。
 - 運算節點必須位於檔案系統所在的相同 AZ 中。
 - 運算節點必須將 [Efa/Enabled](#) 設定為 true。
 - 運算節點必須以 [OnNodeStart](#) 自訂動作 `configure-efa-fsx-lustre-client.sh` 的形式執行組態指令碼。為了方便起見，[FSx 官方文件中](#) 提供的指令碼和公有儲存貯體中提供的指令碼旨在設定運算節點上的 FSx Lustre 用戶端，讓他們使用 EFA。

2. 建立叢集組態檔案 `config.yaml`：

```
Region: region
Image:
  Os: ubuntu2204
HeadNode:
  InstanceType: c5.xlarge
  Networking:
    SubnetId: subnet-xxxxxxxxxx
    AdditionalSecurityGroups:
      - sg-client
  Ssh:
    KeyName: my-ssh-key
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: q1
      ComputeResources:
        - Name: cr1
          Instances:
            - InstanceType: g6.16xlarge
          MinCount: 1
          MaxCount: 3
          Efa:
            Enabled: true
          Networking:
            SubnetIds:
              - subnet-xxxxxxxxxx # Subnet in the same AZ where the file system is
```

```
AdditionalSecurityGroups:
  - sg-client
PlacementGroup:
  Enabled: false
CustomActions:
  OnNodeStart:
    Script: https://us-east-1-aws-parallelcluster.s3.us-east-1.amazonaws.com/
scripts/fsx-lustre-efa/configure-efa-fsx-lustre-client.sh
SharedStorage:
  - MountDir: /fsx
    Name: my-fsxlustre-efa-external
    StorageType: FsxLustre
    FsxLustreSettings:
      FileSystemId: fs-id
```

然後使用該組態建立叢集：

```
pcluster create-cluster \
  --cluster-name fsx-efa-tutorial \
  --cluster-configuration config.yaml \
  --region region
```

使用 EFA 驗證 FSx 正在運作

若要驗證 Lustre 網路流量是否使用 EFA，請使用 `Lustre lnetctl` 工具來顯示指定網路介面的網路流量。為此，請在運算節點中執行下列命令：

```
# Take note of the number of packets flowing through the interface,
# which are specified in statistics:send_count and statistics:recv_count
sudo lnetctl net show --net efa -v

# Generate traffic to the file system
echo 'Hello World' > /fsx/hello-world.txt

# Take note of the number of packets flowing through the interface,
# which are specified in statistics:send_count and statistics:recv_count
sudo lnetctl net show --net efa -v
```

如果功能正常運作，流經界面的封包數量預期會增加。

使用 p6e-gb200 執行個體支援 NVIDIA-Imex

本教學課程說明如何在 P6e-GB200 AWS ParallelCluster 上開始使用，以利用最高 GPU 效能進行 AI 訓練和推論。[p6e-gb200.36xlarge 執行個體只能透過 P6e-GB200 UltraServers 使用](#)，其中 u-p6e-gb200x72 是 Ultraserver Size，而 p6e-gb200.36xlarge 是 [InstanceType](#)，其會形成超伺服器。購買 Ultraserver 時 u-p6e-gb200x72，可透過具有 18 個 p6e-gb200.36xlarge 執行個體的 [適用於 ML 的 EC2 容量區塊](#) 取得。若要進一步了解，請參閱 [P6e-GB200](#)。

AWS ParallelCluster 3.14.0 版：

- 提供此執行個體類型所需的完整 NVIDIA 軟體堆疊（驅動程式、CUDA、EFA、NVIDIA-IMEX）
- 為 P6e-GB200 Ultraserver 建立 nvidia-imex 組態
- 啟用和啟動 P6e-GB200 Ultraserver nvidia-imex 的服務
- 設定 Slurm Block 拓撲外掛程式，讓每個 P6e-GB200 Ultraserver (EC2 容量區塊) 都是大小正確的 Slurm Block（請參閱 3.14.0 版 [版本備註和文件歷史記錄](#) 的項目）。

不過，透過 NVLink 的 GPU-to-GPU 通訊需要額外的組態，特別是包含 ParallelCluster 不會自動產生之 IMEX 網域中運算節點 IP 地址 [nodes_config.cfg](#) 的檔案。為了協助產生此檔案，我們提供 Prolog 指令碼，可自動探索運算節點 IPs，並設定 [nodes_config.cfg](#) 下列 [NVIDIA IMEX Slurm 任務排程器整合建議](#)。本教學課程會逐步解說如何建立 prolog 指令碼、透過 HeadNode 自訂動作部署，以及驗證 IMEX 設定。

Note

從 Amazon Linux 2023、Ubuntu 22.04 和 Ubuntu 24.04 的 v3.14.0 開始，支援 P6e-GB200。AWS ParallelCluster 如需詳細的軟體版本和更新的支援分佈清單，請參閱 [AWS ParallelCluster 變更日誌](#)。

建立 Prolog 指令碼以管理 NVIDIA-Imex

限制：

- 此 prolog 指令碼將在提交專屬任務時執行。這是為了確保 IMEX 重新啟動不會中斷屬於 IMEX 網域之 p6e-Gb200 節點上的任何執行中任務。

以下是您應該在 Slurm 中設定為 Prolog 的 `91_nvidia_imex_prolog.sh` 指令碼。它用於自動更新運算節點上的 `nvidia-imex` 組態。指令碼名稱的字首為 91，以遵循 [SchedMD 的命名慣例](#)。這可確保它在序列中的任何其他 prolog 指令碼之前執行。指令碼會在任務啟動時重新設定 NVIDIA Imex 節點的組態，並重新載入必要的 NVIDIA 協助程式。

Note

如果相同節點上同時啟動多個任務，則不會執行此指令碼，因此我們建議在提交時使用 `--exclusive` 旗標。

```
#!/usr/bin/env bash

# This prolog script configures the NVIDIA IMEX on compute nodes involved in the job
# execution.
#
# In particular:
# - Checks whether the job is executed exclusively.
#   If not, it exits immediately because it requires jobs to be executed exclusively.
# - Checks if it is running on a p6e-gb200 instance type.
#   If not, it exits immediately because IMEX must be configured only on that instance
#   type.
# - Checks if the IMEX service is enabled.
#   If not, it exits immediately because IMEX must be enabled to get configured.
# - Creates the IMEX default channel.
#   For more information about IMEX channels, see https://docs.nvidia.com/multi-node-nvlink-systems/imex-guide/imexchannels.html
# - Writes the private IP addresses of compute nodes into /etc/nvidia-imex/nodes_config.cfg.
# - Restarts the IMEX system service.
#
# REQUIREMENTS:
# - This prolog assumes to be run only with exclusive jobs.

LOG_FILE_PATH="/var/log/parallelcluster/nvidia-imex-prolog.log"
SCONTROL_CMD="/opt/slurm/bin/scontrol"
IMEX_START_TIMEOUT=60
IMEX_STOP_TIMEOUT=15
ALLOWED_INSTANCE_TYPES="^(p6e-gb200)"
IMEX_SERVICE="nvidia-imex"
IMEX_NODES_CONFIG="/etc/nvidia-imex/nodes_config.cfg"
```

```
function info() {
    echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [INFO] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function warn() {
    echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [WARN] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function error() {
    echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [ERROR] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function error_exit() {
    error "$1" && exit 1
}

function prolog_end() {
    info "PROLOG End JobId=${SLURM_JOB_ID}: $0"
    info "-----"
    exit 0
}

function get_instance_type() {
    local token=$(curl -X PUT -s "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
    curl -s -H "X-aws-ec2-metadata-token: ${token}" http://169.254.169.254/latest/meta-data/instance-type
}

function return_if_unsupported_instance_type() {
    local instance_type=$(get_instance_type)

    if [[ ! ${instance_type} =~ ${ALLOWED_INSTANCE_TYPES} ]]; then
        info "Skipping IMEX configuration because instance type ${instance_type} does not support it"
        prolog_end
    fi
}

function return_if_imex_disabled() {
    if ! systemctl is-enabled "${IMEX_SERVICE}" &>/dev/null; then
        warn "Skipping IMEX configuration because system service ${IMEX_SERVICE} is not enabled"
        prolog_end
    fi
}
```



```
fi
}

function return_if_job_is_not_exclusive() {
  if [[ "${SLURM_JOB_OVERSUBSCRIBE}" =~ ^(NO|TOP0)$ ]]; then
    info "Job is exclusive, proceeding with IMEX configuration"
  else
    info "Skipping IMEX configuration because the job is not exclusive"
    prolog_end
  fi
}

function get_ips_from_node_names() {
  local _nodes=$1
  ${SCONTROL_CMD} -ao show node "${_nodes}" | sed 's/^.* NodeAddr=\([^ ]*\).*\1/'
}

function get_compute_resource_name() {
  local _queue_name_prefix=$1
  local _slurmd_node_name=$2
  echo "${_slurmd_node_name}" | sed -E "s/${_queue_name_prefix}(.)-[0-9]+$/\1/"
}

function reload_imex() {
  info "Stopping IMEX"
  timeout ${IMEX_STOP_TIMEOUT} systemctl stop ${IMEX_SERVICE}
  pkill -9 ${IMEX_SERVICE}

  info "Restarting IMEX"
  timeout ${IMEX_START_TIMEOUT} systemctl start ${IMEX_SERVICE}
}

function create_default_imex_channel() {
  info "Creating IMEX default channel"
  MAJOR_NUMBER=$(cat /proc/devices | grep nvidia-caps-imex-channels | cut -d' ' -f1)
  if [ ! -d "/dev/nvidia-caps-imex-channels" ]; then
    sudo mkdir /dev/nvidia-caps-imex-channels
  fi

  # Then check and create device node
  if [ ! -e "/dev/nvidia-caps-imex-channels/channel0" ]; then
    sudo mknod /dev/nvidia-caps-imex-channels/channel0 c $MAJOR_NUMBER 0
    info "IMEX default channel created"
  else

```

```
    info "IMEX default channel already exists"
  fi
}

{
  info "PROLOG Start JobId=${SLURM_JOB_ID}: $0"

  return_if_job_is_not_exclusive
  return_if_unsupported_instance_type
  return_if_imex_disabled

  create_default_imex_channel

  IPS_FROM_CR=$(get_ips_from_node_names "${SLURM_NODELIST}")

  info "Node Names: ${SLURM_NODELIST}"
  info "Node IPs: ${IPS_FROM_CR}"
  info "IMEX Nodes Config: ${IMEX_NODES_CONFIG}"

  info "Updating IMEX nodes config ${IMEX_NODES_CONFIG}"
  echo "${IPS_FROM_CR}" > "${IMEX_NODES_CONFIG}"
  reload_imex

  prolog_end
} 2>&1 | tee -a "${LOG_FILE_PATH}" | logger -t "91_nvidia_imex_prolog"
```

建立 HeadNode OnNodeStart 自訂動作指令碼

建立 `install_custom_action.sh` 自訂動作，其將在運算節點存取的共用目錄中下載上述 `prolog` 指令碼 `/opt/slurm/etc/scripts/prolog.d/`，並設定要執行的適當許可。

```
#!/bin/bash
set -e

echo "Executing $0"

PROLOG_NVIDIA_IMEX=/opt/slurm/etc/scripts/prolog.d/91_nvidia_imex_prolog.sh
aws s3 cp "s3://<Bucket>/91_nvidia_imex_prolog.sh" "${PROLOG_NVIDIA_IMEX}"
chmod 0755 "${PROLOG_NVIDIA_IMEX}"
```

建立叢集

建立包含 P6e-GB200 執行個體的叢集。您可以在下面找到包含 Ultraserer 類型的 SlurmQueues 的範例組態 `u-p6e-gb200x72`。

P6e-GB200 目前僅適用於 Local Zones。有些 [Local Zones 不支援 NAT Gateway](#)，因此請遵循 [Local Zones 的連線選項](#)，因為 ParallelCluster [設定受限環境的安全群組](#) 需要連線到 AWS 服務。請遵循 [使用容量區塊 \(CB\) 啟動執行個體](#) (AWS ParallelClusterLaunch)，因為 Ultraserer 僅供容量區塊使用。

```
HeadNode:
  CustomActions:
    OnNodeStart:
      Script: s3://<s3-bucket-name>/install_custom_action.sh
    S3Access:
      - BucketName: <s3-bucket-name>
  InstanceType: <HeadNode-instance-type>
  Networking:
    SubnetId: <subnet-abcd78901234567890>
  Ssh:
    KeyName: <Key-name>
Image:
  Os: ubuntu2404
Scheduling:
  Scheduler: slurm
  SlurmSettings:
    CustomSlurmSettings:
      - PrologFlags: "Alloc,NoHold"
      - MessageTimeout: 240
  SlurmQueues:
    - CapacityReservationTarget:
        CapacityReservationId: <cr-123456789012345678>
      CapacityType: CAPACITY_BLOCK
      ComputeResources: ### u-p6e-gb200x72
        - DisableSimultaneousMultithreading: true
      Efa:
        Enabled: true
      InstanceType: p6e-gb200.36xlarge
      MaxCount: 18
      MinCount: 18
      Name: cr1
    Name: q1
  Networking:
    SubnetIds:
```

```
- <subnet-1234567890123456>
```

驗證 IMEX 設定

當您提交 Slurm 任務時，將會執行 `91_nvidia_imex_prolog.sh prolog`。以下是檢查 NVIDIA-imex 網域狀態的範例任務。

```
#!/bin/bash
#SBATCH --job-name=nvidia-imex-status-job
#SBATCH --ntasks-per-node=1
#SBATCH --output=slurm-%j.out
#SBATCH --error=slurm-%j.err

QUEUE_NAME="q1"
COMPUTE_RES_NAME="cr1"
IMEX_CONFIG_FILE="/opt/parallelcluster/shared/nvidia-imex/config_${QUEUE_NAME}_
${COMPUTE_RES_NAME}.cfg"

srun bash -c "/usr/bin/nvidia-imex-ctl -N -c ${IMEX_CONFIG_FILE} > result_\
${SLURM_JOB_ID}_\$(hostname).out 2> result_\${SLURM_JOB_ID}_\$(hostname).err"
```

檢查任務的輸出：

```
Connectivity Table Legend:
I - Invalid - Node wasn't reachable, no connection status available
N - Never Connected
R - Recovering - Connection was lost, but clean up has not yet been triggered.
D - Disconnected - Connection was lost, and clean up has been triggered.
A - Authenticating - If GSSAPI enabled, client has initiated mutual authentication.
!V! - Version mismatch, communication disabled.
!M! - Node map mismatch, communication disabled.
C - Connected - Ready for operation
```

```
5/12/2025 06:08:10.580
```

```
Nodes:
```

```
Node #0 - 172.31.48.81 - READY - Version: 570.172
Node #1 - 172.31.48.98 - READY - Version: 570.172
Node #2 - 172.31.48.221 - READY - Version: 570.172
Node #3 - 172.31.49.228 - READY - Version: 570.172
Node #4 - 172.31.50.39 - READY - Version: 570.172
Node #5 - 172.31.50.44 - READY - Version: 570.172
Node #6 - 172.31.51.66 - READY - Version: 570.172
```

```

Node #7 - 172.31.51.157 - READY - Version: 570.172
Node #8 - 172.31.52.239 - READY - Version: 570.172
Node #9 - 172.31.53.80 - READY - Version: 570.172
Node #10 - 172.31.54.95 - READY - Version: 570.172
Node #11 - 172.31.54.183 - READY - Version: 570.172
Node #12 - 172.31.54.203 - READY - Version: 570.172
Node #13 - 172.31.54.241 - READY - Version: 570.172
Node #14 - 172.31.55.59 - READY - Version: 570.172
Node #15 - 172.31.55.187 - READY - Version: 570.172
Node #16 - 172.31.55.197 - READY - Version: 570.172
Node #17 - 172.31.56.47 - READY - Version: 570.172
    
```

Nodes	From\To	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
1		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
2		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
3		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
4		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
5		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
6		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
7		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
8		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
9		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
10		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
11		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
12		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
13		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
14		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
15		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
16		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
17		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

Domain State: UP

使用啟動範本覆寫自訂運算節點網路介面

從 AWS ParallelCluster 3.15.0 開始，`LaunchTemplateOverrides` 參數可讓您使用參考啟動範本中的組態覆寫預設網路介面組態，以自訂運算節點的網路介面。用於覆寫的啟動範本的網路介面區段會覆寫運算節點的整個網路介面區段。

本教學課程會逐步解說覆寫 p6-b300.48xlarge 運算節點預設網路組態的範例。當您需要與預設 AWS ParallelCluster 設定不同的特定網路介面組態時，此自訂很有用。在此範例中，我們為 P6-B300 執行個體設定使用案例 2，如 [Amazon EC2 EFA 支援的執行個體類型文件](#) 所述。

Note

建議您使用 AWS CLI 來建立啟動範本，而非主控台，以獲得最大的彈性。

Note

啟動範本應該只包含網路介面覆寫。AWS ParallelCluster 具有防止覆寫其他參數的驗證。

Warning

如果您使用覆寫以使用執行個體類型不支援的方式設定網路介面，則執行個體將無法啟動。

先決條件

- AWS ParallelCluster [已安裝](#) 3.15.0 版或更新版本。
- AWS CLI [已安裝並設定](#)。
- 您有一個 IAM 角色，具有執行 CLI [pcluster](#) 所需的[許可](#)。

步驟 1：建立安全群組

建立要在覆寫中使用的啟動範本時，您必須參考安全群組。在叢集建立之前，運算資源的預設 AWS ParallelCluster 安全群組不存在，因此您必須建立自訂安全群組。然後，前端節點安全群組必須參考此安全群組，以允許前端節點和運算節點之間的流量。

如果您要更新現有叢集以自訂新容量，則可以在啟動範本中使用預設 AWS ParallelCluster 運算節點安全群組，而不是建立自訂叢集。

建立下列兩個安全群組：

- 標頭節點額外安全群組 (sg-1234abcd)：
 - 傳入：來自運算安全群組的所有流量
- 運算安全群組 (sg-abcd1234)：
 - 輸入：來自前端節點安全群組的所有流量
 - 輸入：來自自我的所有流量 compute-to-compute)
 - 輸出：預設全部允許

步驟 2：建立啟動範本

建立啟動範本，定義 p6-b300.48xlarge 運算節點的網路介面組態。對於主要網路界面（網路卡索引 0、裝置索引 0），請使用 ENA（預設）網路界面。對於剩餘的網路卡，請建立僅限 EFA 的介面（網路卡索引 1-16，裝置索引 0）和 ENA（預設）介面（網路卡索引 1-16，裝置索引 1）。

執行下列 AWS CLI 命令來建立啟動範本 (lt-123456789)：

```
aws ec2 create-launch-template \  
  --region us-east-1 \  
  --launch-template-name override-lt \  
  --launch-template-data '{  
    "NetworkInterfaces": [  
      {"NetworkCardIndex":0, "DeviceIndex":0, "Groups":["sg-abcd1234"],  
"SubnetId":"subnet-123456789"},  
      {"NetworkCardIndex":1, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":  
["sg-abcd1234"], "SubnetId":"subnet-123456789"},  
      {"NetworkCardIndex":1, "DeviceIndex":1, "Groups":["sg-abcd1234"],  
"SubnetId":"subnet-123456789"},  
      {"NetworkCardIndex":2, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":  
["sg-abcd1234"], "SubnetId":"subnet-123456789"},  
      {"NetworkCardIndex":2, "DeviceIndex":1, "Groups":["sg-abcd1234"],  
"SubnetId":"subnet-123456789"},  
      {"NetworkCardIndex":3, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":  
["sg-abcd1234"], "SubnetId":"subnet-123456789"},  
      {"NetworkCardIndex":3, "DeviceIndex":1, "Groups":["sg-abcd1234"],  
"SubnetId":"subnet-123456789"},  
    ]  
  }'
```

```
    {"NetworkCardIndex":4, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":4, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":5, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":5, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":6, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":6, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":7, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":7, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":8, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":8, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":9, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":9, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":10, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":10, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":11, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":11, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":12, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":12, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":13, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":13, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":14, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":14, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
```



```

    {"NetworkCardIndex":15, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":15, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":16, "DeviceIndex":0, "InterfaceType":"efa-only", "Groups":
["sg-abcd1234"], "SubnetId":"subnet-123456789"},
    {"NetworkCardIndex":16, "DeviceIndex":1, "Groups":["sg-abcd1234"],
"SubnetId":"subnet-123456789"}
  ]
}'

```

步驟 3：建立具有啟動範本覆寫的叢集

建立使用 `LaunchTemplateOverrides` 參數來參考您建立的啟動範本的叢集組態。

```

Region: us-east-1
HeadNode:
  InstanceType: c5.xlarge
  Networking:
    SubnetId: subnet-abcdefghi
    AdditionalSecurityGroups:
      # Add the head node SG that allows traffic from the compute node SG
      - sg-1234abcd
  ...

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
    Networking:
      SubnetIds:
        - subnet-123456789
  ComputeResources:
    - Name: compute-resource1
      InstanceType: p6-b300.48xlarge
      Efa:
        Enabled: false # The override replaces all network interface configuration,
so this setting is ignored
      LaunchTemplateOverrides:
        LaunchTemplateId: lt-123456789
        Version: 1 # If the launch template is updated, then the new version should
be specified here.

```

AWS ParallelCluster 故障診斷

下列各節提供使用時可能發生之問題的疑難排解秘訣 AWS ParallelCluster。AWS ParallelCluster 社群會維護 Wiki 頁面，在 [AWS ParallelCluster GitHub Wiki](#) 上提供許多疑難排解秘訣。如需已知問題的清單，請參閱 [已知問題](#)。

主題

- [嘗試建立叢集](#)
- [嘗試執行任務](#)
- [嘗試更新叢集](#)
- [嘗試存取儲存體](#)
- [嘗試刪除叢集](#)
- [嘗試升級 AWS ParallelCluster API 堆疊](#)
- [查看運算節點初始化中的錯誤](#)
- [叢集運作狀態指標疑難排解](#)
- [對叢集部署問題進行故障診斷](#)
- [使用 Terraform 對叢集部署進行故障診斷](#)
- [對擴展問題進行故障診斷](#)
- [置放群組和執行個體啟動問題](#)
- [取代目錄](#)
- [對 Amazon DCV 中的問題進行故障診斷](#)
- [使用 AWS Batch 整合對叢集中的問題進行故障診斷](#)
- [對與 Active Directory 的多使用者整合進行故障診斷](#)
- [對自訂 AMI 問題進行故障診斷](#)
- [在 cfn-hup 未執行時對叢集更新逾時進行故障診斷](#)
- [網路故障診斷](#)
- [onNodeUpdated 自訂動作上的叢集更新失敗](#)
- [使用自訂 Slurm 組態查看錯誤](#)
- [叢集警示](#)
- [解決導致錯誤或失敗的作業系統組態變更](#)

嘗試建立叢集

使用 3.5.0 版和更新 AWS ParallelCluster 版本來建立叢集時，如果叢集建立失敗，並將 `--rollback-on-failure` 設為 `false`，請使用 [pcluster describe-cluster](#) CLI 命令來取得狀態和失敗資訊。在此情況下，`pcluster describe-cluster` 輸出 `clusterStatus` 的預期值為 `CREATE_FAILED`。檢查輸出中的 `failures` 區段以尋找 `failureCode` 和 `failureReason`。然後，在下一節中，尋找其他故障診斷說明 `failureCode` 的相符項目。如需詳細資訊，請參閱 [pcluster describe-cluster](#)。

在下列各節中，我們建議您檢查前端節點上的日誌，例如 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案。如需 AWS ParallelCluster 日誌以及如何檢視它們的詳細資訊，請參閱 [除錯的金鑰日誌](#) 和 [擷取和保留日誌](#)。

如果您沒有 `failureCode`，請導覽至 CloudFormation 主控台以檢視叢集堆疊。檢查 `StatusReason` 其他資源上的 `HeadNodeWaitCondition` 或 失敗，以尋找其他失敗詳細資訊。如需詳細資訊，請參閱 [在上檢視 CloudFormation 事件 CREATE_FAILED](#)。檢查前端節點上的 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案。如果叢集建立因為前端節點建立失敗而失敗，且叢集日誌無法在叢集日誌群組中使用，您必須在失敗時保留叢集，指定 `--rollback-on-failure = True` 並從前端節點本身擷取日誌。

failureCode 是 OnNodeConfiguredExecutionFailure

- 為什麼失敗？

您已在組態中前端節點區段 `OnNodeConfigured` 的 `Commands` 中提供自訂指令碼，以建立叢集。不過，自訂指令碼無法執行。

- 如何解決？

檢查 `/var/log/cfn-init.log` 檔案以進一步了解故障，以及如何修正自訂指令碼中的問題。在此日誌結尾附近，您可能會在 `Running command runpostinstall` 訊息之後看到與 `OnNodeConfigured` 指令碼相關的執行資訊。

failureCode 是 OnNodeConfiguredDownloadFailure

- 為什麼失敗？

您已在組態中前端節點區段 `OnNodeConfigured` 的 `Commands` 中提供自訂指令碼，以建立叢集。不過，自訂指令碼無法下載。

- 如何解決？

請確定 URL 有效且已正確設定存取。如需自訂引導指令碼組態的詳細資訊，請參閱 [自訂引導操作](#)。

檢查 `/var/log/cfn-init.log` 檔案。在接近此日誌結尾處，您可能會在Running command `runpostinstall` 訊息之後看到與OnNodeConfigured指令碼處理相關的執行資訊，包括下載。

failureCode 是 OnNodeConfiguredFailure

- 為什麼失敗？

您已在組態中前端節點區段OnNodeConfigured的 中提供自訂指令碼，以建立叢集。不過，在叢集部署中使用自訂指令碼失敗。無法判斷立即原因，且需要額外調查。

- 如何解決？

檢查 `/var/log/cfn-init.log` 檔案。在接近此日誌結尾處，您可能會在Running command `runpostinstall` 訊息後看到OnNodeConfigured指令碼處理的相關資訊。

failureCode 是 OnNodeStartExecutionFailure

- 為什麼失敗？

您已在組態中前端節點區段OnNodeStart的 中提供自訂指令碼，以建立叢集。不過，自訂指令碼無法執行。

- 如何解決？

檢查 `/var/log/cfn-init.log` 檔案以進一步了解故障，以及如何修正自訂指令碼中的問題。在此日誌結尾附近，您可能會在Running command `runpreinstall` 訊息之後看到與OnNodeStart指令碼相關的執行資訊。

failureCode 是 OnNodeStartDownloadFailure

- 為什麼失敗？

您已在組態中前端節點區段OnNodeStart的 中提供自訂指令碼，以建立叢集。不過，自訂指令碼無法下載。

- 如何解決？

請確定 URL 有效且已正確設定存取。如需自訂引導指令碼組態的詳細資訊，請參閱 [自訂引導操作](#)。

檢查 `/var/log/cfn-init.log` 檔案。在接近此日誌結尾處，您可能會在 `Running command runpreinstall` 訊息之後看到與 `OnNodeStart` 指令碼處理相關的執行資訊，包括下載。

failureCode 是 OnNodeStartFailure

- 為什麼失敗？

您在組態中前端節點區段 `OnNodeStart` 的中提供自訂指令碼，以建立叢集。不過，在叢集部署中使用自訂指令碼失敗。無法判斷立即原因，且需要額外調查。

- 如何解決？

檢查 `/var/log/cfn-init.log` 檔案。在接近此日誌結尾處，您可能會在 `Running command runpreinstall` 訊息後看到與 `OnNodeStart` 指令碼處理相關的資訊。

failureCode 是 EbsMountFailure

- 為什麼失敗？

叢集組態中定義的 EBS 磁碟區無法掛載。

- 如何解決？

檢查 `/var/log/chef-client.log` 檔案以取得失敗詳細資訊。

failureCode 是 EfsMountFailure

- 為什麼失敗？

叢集組態中定義的 Amazon EFS 磁碟區無法掛載。

- 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 [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)。

檢查 `/var/log/chef-client.log` 檔案以取得失敗詳細資訊。

failureCode 是 FsxMountFailure

- 為什麼失敗？

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 [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)。

檢查 `/var/log/chef-client.log` 檔案以取得失敗詳細資訊。

failureCode 是 RaidMountFailure

- 為什麼失敗？

叢集組態中定義的 RAID 磁碟區無法掛載。

- 如何解決？

檢查 `/var/log/chef-client.log` 檔案以取得失敗詳細資訊。

failureCode 是 AmiVersionMismatch

- 為什麼失敗？

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 堆疊詳細資訊，並檢查 Status Reason 以取得 AWS ParallelCluster 版本和 AMI HeadNodeWaitCondition 的其他詳細資訊。如需詳細資訊，請參閱 [在上檢視 CloudFormation 事件 CREATE_FAILED](#)。

- 如何解決？

確定用於建立自訂 AMI 的 AWS ParallelCluster 版本與用於設定叢集的 AWS ParallelCluster 版本相同。您可以變更自訂 AMI 版本或 pcluster CLI 版本，使其相同。

failureCode 是 InvalidAmi

- 為什麼失敗？

自訂 AMI 無效，因為它不是使用建置的 AWS ParallelCluster。

- 如何解決？

使用 `pcluster build-image` 命令讓 AMI 成為父映像，以建立 AMI。如需詳細資訊，請參閱 [pcluster build-image](#)。

failureCode HeadNodeBootstrapFailure 具有 **failureReason** 無法設定前端節點的。

- 為什麼失敗？

無法判斷立即原因，且需要額外調查。例如，叢集可能處於受保護狀態，這可能是因為無法佈建靜態運算機群所致。

- 如何解決？

檢查 `/var/log/chef-client.log` 檔案以取得失敗詳細資訊。

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`，則叢集處於受保護狀態。如需詳細資訊，請參閱 [如何偵錯受保護模式](#)。

failureCode 是 **failureReason HeadNodeBootstrapFailure**，叢集建立逾時。

- 為什麼失敗？

根據預設，叢集建立需要 30 分鐘的時間限制才能完成。如果叢集建立未在此時間範圍內完成，叢集建立會失敗並出現逾時錯誤。叢集建立可能因不同原因而逾時。例如，逾時失敗可能是由前端節點建立失敗、網路問題、在前端節點中執行時間過長的自訂指令碼、在運算節點中執行的自訂指令碼中的錯誤，或運算節點佈建的等待時間過長所造成。無法判斷立即原因，且需要額外調查。

- 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。如需日誌以及如何取得它們的詳細資訊 AWS ParallelCluster，請參閱 [除錯的金鑰日誌](#) 和 [擷取和保留日誌](#)。

您可能會在這些日誌中發現以下內容。

- 在的結尾 **Waiting for static fleet capacity provisioning** 附近看到 **chef-client.log**

這表示等待靜態節點啟動時，叢集建立逾時。如需詳細資訊，請參閱 [查看運算節點初始化中的錯誤](#)。

- 查看 **OnNodeConfigured** 或 **OnNodeStart** 前端節點指令碼尚未在結尾完成 **cfn-init.log**

這表示 **OnNodeConfigured** 或 **OnNodeStart** 自訂指令碼需要很長時間才能執行，並導致逾時錯誤。檢查您的自訂指令碼是否有可能導致其長時間執行的問題。如果您的自訂指令碼需要很長的時間來執行，請考慮將 `DevSettings` 區段新增至叢集組態檔案來變更逾時限制，如下列範例所示：

```
DevSettings:
  Timeouts:
    HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- 找不到日誌，或前端節點未成功建立

前端節點可能無法成功建立，也找不到日誌。在這種情況下，您可以檢查 CloudFormation 堆疊事件和前端節點主控台日誌，以取得其他失敗詳細資訊。您可以透過 Amazon EC2 主控台或執行下列 Amazon EC2 CLI 命令來擷取前端節點主控台日誌：

```
aws ec2 get-console-output --instance-id HEAD_NODE_INSTANCE_ID --output text
```

failureCode HeadNodeBootstrapFailure 具有 **failureReason** 無法引導前端節點的。

- 為什麼失敗？

無法判斷立即原因，且需要額外調查。

- 如何解決？

檢查 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案。

failureCode 是 ResourceCreationFailure

- 為什麼失敗？

在叢集建立程序期間，某些資源的建立失敗。失敗可能因為各種原因而發生。例如，資源建立失敗的原因可能是容量問題或設定錯誤的 IAM 政策。

- 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

- 為什麼失敗？

無法判斷立即原因，且需要額外調查。

- 如何解決？

在 CloudFormation 主控台中，檢視叢集堆疊並檢查 Status Reason 以取得 HeadNodeWaitCondition 其他失敗詳細資訊。

檢查 `/var/log/cfn-init.log` 和 `/var/log/chef-client.log` 檔案。

在 CloudFormation 堆疊 WaitCondition timed out... 中查看

如需詳細資訊，請參閱 [failureCode 是 failureReason HeadNodeBootstrapFailure](#)，叢集建立逾時。

在 CloudFormation 堆疊 Resource creation cancelled 中查看

如需詳細資訊，請參閱 [failureCode 是 ResourceCreationFailure](#)。

在 CloudFormation 堆疊中查看 Failed to run cfn-init... 或其他錯誤

如需其他失敗詳細資訊，`/var/log/chef-client.log` 請參閱 `/var/log/cfn-init.log` 和。

查看以 `chef-client.log` 結尾 `INFO: Waiting for static fleet capacity provisioning`

這與等待靜態節點啟動時的叢集建立逾時有關。如需詳細資訊，請參閱[查看運算節點初始化中的錯誤](#)。

查看 `Failed to run preinstall or postinstall in cfn-init.log`

您在叢集組態 `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` 檔案以取得自訂指令碼錯誤詳細資訊。

在 CloudFormation 堆疊 `This AMI was created with xxx, but is trying to be used with xxx...` 中查看

如需詳細資訊，請參閱[failureCode 是 AmiVersionMismatch](#)。

在 CloudFormation 堆疊 `This AMI was not baked by AWS ParallelCluster...` 中查看

如需詳細資訊，請參閱[failureCode 是 InvalidAmi](#)。

查看 `pcluster create-cluster` 命令無法在本機執行

如需失敗詳細資訊，`~/parallelcluster/pcluster-cli.log` 請參閱本機檔案系統中的。

其他支援

請遵循中的疑難排解指引[對叢集部署問題進行故障診斷](#)。

檢查 [GitHub 上的 GitHub 已知問題](#) 是否涵蓋您的案例 AWS ParallelCluster GitHub。

嘗試執行任務

如果您在嘗試執行任務時在 中執行問題，以下章節提供可能的疑難排解解決方案。

`srun` 互動式任務失敗並發生錯誤 `srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf`

- 為什麼失敗？

您執行 `srun` 命令來提交任務，然後使用 `pcluster update-cluster` 命令來增加佇列的大小，而不需在更新完成後重新啟動 Slurm 協助程式。

Slurm 會在樹狀結構中組織 Slurm 協助程式，以最佳化通訊。此階層只會在精靈啟動時更新。

假設您使用 `srun` 啟動任務，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時啟動。然後，會將您的任務排入其中一個新運算節點的 Slurm 佇列。在此情況下，精靈和 Slurm 都不會 `srun` 偵測新的運算節點。會 `srun` 傳回錯誤，因為它不會偵測新的節點。

- 如何解決？

重新啟動所有運算節點上的協助程式，然後使用 Slurm `srun` 提交您的任務。您可以執行重新啟動運算節點的 `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 [控制項重新啟動](#)。您也可以請求重新啟動對應的 `systemd` 服務，以手動重新啟動運算節點上的 Slurm 協助程式。

任務使用 `squeue` 命令卡在 CF 狀態

這可能是動態節點啟動時的問題。如需詳細資訊，請參閱 [查看運算節點初始化中的錯誤](#)。

執行大規模任務並查看 `nfsd: too many open connections, consider increasing the number of threads in /var/log/messages`

透過網路檔案系統，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，因為網路用於寫入聯網和 I/O 指標的資料。

使用第 5 代執行個體時，我們使用 ENA 驅動程式來公開封包計數器。當網路達到執行個體頻寬限制 AWS 時，這些計數器會計算成形的封包。您可以檢查這些計數器，以查看它們是否大於 0。如果是，則您已超過頻寬限制。您可以執行來檢視這些計數器 `ethtool -S eth0 | grep exceeded`。

超過網路限制通常是由於支援太多 NFS 連線所致。這是達到或超過網路限制時要檢查的第一件事之一。

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
contrack_allowance_exceeded: 0
linklocal_allowance_exceeded: 0
```

若要避免收到此訊息，請考慮將前端節點執行個體類型變更為效能更高的執行個體類型。考慮將資料儲存體移至未匯出為 NFS 共用的共用儲存檔案系統，例如 Amazon EFS 或 Amazon FSx。如需詳細資訊，請參閱 [共用儲存](#) 和 GitHub 上 AWS ParallelCluster Wiki 的 [最佳實務](#)。

執行 MPI 任務

啟用偵錯模式

若要啟用 OpenMPI 除錯模式，請參閱 [Open MPI 有哪些控制項有助於除錯](#)。

若要啟用 IntelMPI 除錯模式，請參閱 [其他環境變數](#)。

在任務輸出 **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。若要了解如何從應用程式取得 MPI 除錯日誌，請參閱 [啟用偵錯模式](#)。

此錯誤的可能原因是您的應用程式已針對特定 MPI 實作進行編譯，例如 OpenMPI，而且您嘗試使用不同的 MPI 實作執行它，例如 IntelMPI。請確定您使用相同的 MPI 實作來編譯和執行應用程式。

在停用受管 DNS **mpirun** 的情況下使用

對於使用 [SlurmSettings / Dns / DisableManagedDns](#) 和 [UseEc2Hostnames](#) 設定為 建立的叢集 `true`，DNS 不會解析 Slurm 節點名稱。nodenames 如果未啟用，且 MPI 任務是在 Slurm 內容中執行，則 Slurm 可以引導 MPI 程序。我們建議您遵循 [Slurm MPI 使用者指南](#) 中的指引，以使用執行 MPI 任務 Slurm。

嘗試更新叢集

下節針對您嘗試更新叢集時可能發生的問題，提供可能的疑難排解解決方案。

pcluster update-cluster 命令無法在本機執行

如需失敗詳細資訊，`~/parallelcluster/pcluster-cli.log` 請參閱本機檔案系統中的。

使用 `clusterStatus UPDATE_FAILEDpcluster describe-cluster` 命令查看

根本原因

為了識別失敗的根本原因，起點是查看叢集堆疊事件和前端節點 `/var/log/chef-client.log` 中的。

可能的原因是至少有一個叢集節點未套用更新。您可以在日誌 `/var/log/chef-client.log` 中尋找，以擷取無法在前端節點 `Check cluster readiness` 中更新之節點的清單。

檢查 [GitHub 上的 GitHub 已知問題](#) AWS ParallelCluster 中是否提及您的問題。GitHub

防止

如果叢集中至少有一個節點未成功套用更新，則叢集更新可能會失敗。為了降低叢集更新失敗的風險，建議您在啟動更新之前終止中斷的節點。可能中斷的節點範例是運算節點處於 `COMPLETING` 狀態的時間超過預期的 `epilog` 持續時間。若要偵測這些節點，您可以執行下列命令，根據您的需求調整 `threshold` 值（該值必須大於對 `epilogs` 預期的最長持續時間）。

```
$ scontrol show nodes --json | jq -r --argjson threshold 60 '
  .nodes[] | select(.state | index("COMPLETING")) |
  select((now - .last_busy.number) > $threshold) |
  .name
'
```

復原

如果更新失敗，復原是預期會復原叢集狀態的機制。

如果轉返失敗，叢集狀態不是確定性的。在這種情況下，可能會 `clustermgtd` 停止以防止失敗的擴增。我們建議您在前端節點上執行下列命令來啟動它。將 Python 版本調整為與您的版本一起隨附的 AWS ParallelCluster 版本：

```
$ /opt/parallelcluster/pyenv/versions/3.12.11/envs/cookbook_virtualenv/bin/
supervisorctl start clustermgtd
```

叢集更新逾時

這可能是與cfn-hup未執行相關的問題。如果cfn-hup魔抗被外部原因終止，則不會自動重新啟動。如果cfn-hup未執行，在叢集更新期間，CloudFormation堆疊會如預期啟動更新程序，但前端節點上不會啟動更新程序，且堆疊部署最終會逾時。如需詳細資訊，請參閱 [在 cfn-hup 未執行時對叢集更新逾時進行故障診斷](#) 以疑難排解並從問題中復原。

嘗試存取儲存體

了解嘗試存取儲存體的疑難排解秘訣。

使用外部 Amazon FSx for Lustre 檔案系統

請確定叢集和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 傳入和傳出 TCP 流量的安全群組相關聯。如需如何設定安全群組的詳細資訊，請參閱 [FileSystemId](#)。

使用外部 Amazon Elastic File System 檔案系統

請確定叢集和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 傳入和傳出 TCP 流量的安全群組相關聯。如需如何設定安全群組的詳細資訊，請參閱 [FileSystemId](#)。

嘗試刪除叢集

如果您在嘗試刪除叢集時發生錯誤，以下各節提供常見案例的疑難排解秘訣。

`pcluster delete-cluster` 命令無法在本機執行

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。

檢查您的問題是否在 [GitHub 上的 GitHub 已知問題](#) AWS ParallelCluster 中提及。GitHub

嘗試升級 AWS ParallelCluster API 堆疊

如果您遇到錯誤，例如嘗試升級 AWS ParallelCluster API 堆疊 `UPDATE_FAILED` 時，建議您在 GitHub 上 [AWS ParallelCluster Wiki](#) 的已知問題區段中檢查解決方案。例如，請參閱 [ParallelCluster API Stack Upgrade Fails for ECR 資源](#)，這會識別一個可能的問題並提供緩解選項。

查看運算節點初始化中的錯誤

下列各節提供當您在運算節點初始化中看到錯誤時的疑難排解秘訣。這包括引導錯誤、查看日誌中的錯誤，以及如果沒有任何案例適用於您的特定情況，應前往何處。

主題

- [在 Node bootstrap error 中查看 clustermgtd.log](#)
- [我設定隨需容量保留 ODCRs\) 或區域預留執行個體](#)
- [當我無法執行任務An error occurred \(VcpuLimitExceeded\)slurm_resume.log時查看，或當我無法建立叢集時clustermgtd.log查看。](#)
- [當我無法執行任務An error occurred \(InsufficientInstanceCapacity\)slurm_resume.log時查看，或當我無法建立叢集時clustermgtd.log查看。](#)
- [使用 查看節點處於 DOWN 狀態 Reason \(Code:InsufficientInstanceCapacity\)...](#)
- [在 cannot change locale \(en_US.utf-8\) because it has an invalid name 中查看 slurm_resume.log](#)
- [上述案例都不適用於我的情況](#)

在 Node bootstrap error 中查看 clustermgtd.log

問題與運算節點無法引導有關。如需如何偵錯叢集保護模式問題的資訊，請參閱 [如何偵錯受保護模式](#)。

我設定隨需容量保留 ODCRs) 或區域預留執行個體

包含具有多個網路介面的執行個體ODCRs，例如 P4d, P4de 和 AWS Trainium (Trn)

在叢集組態檔案中，檢查 HeadNode 是否位於公有子網路中，以及運算節點是否位於私有子網路中。

ODCRs是目標 ODCRS

遵循 中的指示，**Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'**。即使我已經/opt/slurm/etc/pcluster/run_instances_overrides.json就地查看 [使用隨需容量預留 \(ODCR\) 啟動執行個體](#)

如果您使用 3.1.1 到 3.2.1 AWS ParallelCluster 版搭配目標 ODCRs，而且您也使用[執行執行個體覆寫 JSON 檔案](#)，則您的 JSON 檔案格式可能不正確。您可以在 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.
```

執行下列動作來驗證 JSON 檔案格式是否正確：

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

在叢集建立失敗 **Found RunInstances parameters override.clustermgtd.log** 時查看，或在執行任務失敗 **slurm_resume.log** 時查看

如果您使用的是 [執行執行個體覆寫 JSON 檔案](#)，請檢查您是否在 `/opt/slurm/etc/pcluster/run_instances_overrides.json` 檔案中正確設定佇列名稱和運算資源名稱。

當我無法執行任務 **An error occurred**

(**InsufficientInstanceCapacity**) **slurm_resume.log** 時查看，或當我無法建立叢集 **clustermgtd.log** 時查看。

使用 PG-ODCR (配置群組 ODCR)

使用相關聯的置放群組建立 ODCR 時，必須在組態檔案中使用相同的置放群組名稱。在叢集組態中設定對應的 [置放群組名稱](#)。

使用區域預留執行個體

如果您在叢集組態 `true` 中使用區域預留執行個體搭配 `PlacementGroup / Enabled` 到，您可能會看到錯誤，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.
```

您可能會看到這種情況，因為區域預留執行個體未放置在相同的 UC (或脊椎) 中，這可能會在使用置放群組時導致容量不足錯誤 (ICEs)。您可以在叢集組態中停用 `PlacementGroup` 群組設定，以判斷叢集是否可以配置執行個體，藉此檢查此案例。

當我無法執行任務An error occurred (VcpuLimitExceeded)slurm_resume.log時查看，或當我無法建立叢集時clustermgtd.log查看。

檢查您帳戶上的 vCPU 限制，了解您使用的特定 Amazon EC2 執行個體類型。如果您看到的 vCPUs 比您請求的 vCPU 為零或更少，請請求提高限制。如需有關如何檢視目前限制和請求新限制的資訊，請參閱 [《Amazon EC2 使用者指南》中的 Amazon EC2 服務配額](#)。Amazon EC2

當我無法執行任務An error occurred (InsufficientInstanceCapacity)slurm_resume.log時查看，或當我無法建立叢集時clustermgtd.log查看。

您遇到容量不足的問題。請遵循 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> : //。

使用查看節點處於 DOWN 狀態 Reason (Code:InsufficientInstanceCapacity)...

您遇到容量不足的問題。請遵循 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> : //。如需 AWS ParallelCluster 快速容量不足容錯移轉模式的詳細資訊，請參閱 [Slurm 叢集快速容量不足容錯移轉](#)。

在 cannot change locale (en_US.utf-8) because it has an invalid name 中查看 slurm_resume.log

如果您的 yum 安裝程序失敗，使得地區設定處於不一致狀態，就會發生這種情況。例如，這可能會在使用者終止安裝程序時造成。

若要驗證原因，請採取下列動作：

- 執行 `su - pcluster-admin`。
shell 會顯示錯誤，例如 `cannot change locale...no such file or directory`。
- 執行 `localedef --list`。
傳回空白清單，或不包含預設地區設定。

- 使用 `yum history` 和 檢查最後一個 `yum` 命令 `yum history info #ID`。最後一個 ID 是否具有 `Return-Code: Success` ?

如果最後一個 ID 沒有 `Return-Code: Success` , 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 重建地區設定 `yum reinstall glibc-all-langpacks`。重建之後, 如果問題已修正, `su - pcluster-admin` 不會顯示錯誤或警告。

上述案例都不適用於我的情況

若要疑難排解運算節點初始化問題, 請參閱 [故障診斷節點初始化問題](#)。

檢查 [GitHub 上的 GitHub 已知問題](#) 是否涵蓋您的案例 AWS ParallelCluster GitHub 。

叢集運作狀態指標疑難排解

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。在下列各節中, 您可以了解儀表板運作狀態指標, 以及疑難排解和解決問題時可採取的動作。

主題

- [請參閱執行個體佈建錯誤圖表](#)
- [查看運作狀態不佳的執行個體錯誤圖表](#)
- [請參閱運算機群閒置時間圖表](#)

請參閱執行個體佈建錯誤圖表

如果您在 `Instance Provisioning Errors` 圖形中看到非零值, 則表示用於備份 slurm 節點的 Amazon EC2 執行個體無法在 `CreateFleet` 或 `RunInstance` API 上啟動。

查看 **IAMPolicyErrors**

- 發生了什麼事?

許多執行個體無法啟動, 這是由於錯誤碼的許可不足所致 `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` 檔案是否有靜態節點錯誤詳細資訊。檢查 `slurm_resume.log` 檔案以取得動態節點錯誤詳細資訊。使用詳細資訊進一步了解必須新增的缺少許可。

查看 VcpuLimitErrors

- 發生了什麼事？

AWS ParallelCluster 無法啟動執行個體，因為它已達到您 AWS 帳戶 為叢集運算節點設定之特定 Amazon EC2 執行個體類型的 上的 vCPU 限制。

- 如何解決？

檢查 `clustermgtd` 檔案中是否有靜態節點的 `VcpuLimitExceeded` 錯誤，並檢查 `slurm_resume.log` 檔案中是否有動態節點以取得其他詳細資訊。若要解決此問題，您可以請求提高 vCPU 限制。如需如何檢視目前限制和請求新限制的詳細資訊，請參閱 [《Amazon Elastic Compute Cloud Linux 執行個體使用者指南》](#) 中的 [Amazon Elastic Compute Cloud 服務配額](#)。

查看 VolumeLimitErrors

- 發生了什麼事？

您已達到的 Amazon EBS 磁碟區限制 AWS 帳戶，AWS ParallelCluster 且無法使用錯誤代碼 `InsufficientVolumeCapacity` 或 啟動執行個體 `VolumeLimitExceeded`。

- 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 AWS 區域清除現有磁碟區，或聯絡 AWS Support Center 提交請求以增加 Amazon EBS 磁碟區限制。

查看 InsufficientCapacityErrors

- 發生了什麼事？

AWS ParallelCluster 沒有足夠的容量來啟動 Amazon EC2 執行個體以恢復節點。

- 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，以取得容量不足的錯誤詳細資訊。若要對問題進行疑難排解，請遵循 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> 上的指引。

OtherInstanceLaunchFailures

- 發生了什麼事？

用於備份運算節點的 Amazon EC2 執行個體無法使用 `CreateFleet` 或 `RunInstance` API 啟動。

- 如何解決？

檢查 `clustermgtd` 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錯誤詳細資訊。

查看運作狀態不佳的執行個體錯誤圖表

- 發生了什麼事？

許多運算執行個體已啟動，但後來因運作狀態不佳而終止。

- 如何解決？

如需故障診斷運作狀態不佳節點的詳細資訊，請參閱 [對非預期的節點替換和終止進行故障診斷](#)。

查看 InstanceBootstrapTimeoutError

- 發生了什麼事？

執行個體無法在 `resume_timeout` (適用於動態節點) 或 `node_replacement_timeout` (適用於靜態節點) 中連結叢集。如果未針對運算節點正確設定網路，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需要太長的時間才能完成，則可能會發生這種情況。

- 如何解決？

對於動態節點，請檢查 `clustermgtd` 日誌 (`/var/log/parallelcluster/clustermgtd`) 是否有運算節點 IP 地址和錯誤，如下所示：

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

對於靜態節點，請檢查 `clustermgtd` 日誌 (`/var/log/parallelcluster/clustermgtd`) 是否有運算節點 IP 地址和錯誤，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` 日誌檔案擷取有問題的運算節點 IP 地址。

查看 `EC2HealthCheckErrors`

- 發生了什麼事？

執行個體未通過 Amazon EC2 運作狀態檢查。

- 如何解決？

如需如何對此問題進行故障診斷的資訊，請參閱[對狀態檢查失敗的執行個體進行故障診斷](#)。

查看 `ScheduledEventHealthCheckErrors`

- 發生了什麼事？

執行個體未通過 Amazon EC2 排程事件運作狀態檢查，而且運作狀態不佳。

- 如何解決？

如需如何疑難排解此問題的資訊，請參閱[執行個體的排程事件](#)。

查看 `NoCorrespondingInstanceErrors`

- 發生了什麼事？

AWS ParallelCluster 找不到執行個體後端節點。節點可能會在引導操作期間自行終止。[/CustomActions/ OnNodeStart](#) | [SlurmQueues OnNodeConfigured](#) 指令碼或網路錯誤可能會產生 `NoCorrespondingInstanceErrors`。

- 如何解決？

如需其他詳細資訊，請檢查運算節點 `/var/log/cloud-init-output.log` 的。

請參閱運算機群閒置時間圖表

看到MaxDynamicNodeIdleTime明顯長於閒置時間縮減閾值的

- 發生了什麼事？

您的執行個體未正確終止。MaxDynamicNodeIdleTime顯示動態節點由 Amazon EC2 執行個體支援的閒置時間上限，以秒為單位。閒置時間縮減閾值衍生自叢集組態 [ScaledownIdletime](#) 參數。當運算節點閒置超過閒置時間縮放秒時，會關閉節點Slurm電源並 AWS ParallelCluster 終止備份執行個體。在這種情況下，某些情況會阻止執行個體終止。

- 如何解決？

如需此問題的詳細資訊，請參閱 [取代、終止或關閉有問題的執行個體和節點](#) 中的 [對擴展問題進行故障診斷](#)。

對叢集部署問題進行故障診斷

如果您的叢集無法建立並復原堆疊建立，您可以查看日誌檔案來診斷問題。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": "3.15.0",
    "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

- [在上檢視 CloudFormation 事件 CREATE_FAILED](#)
- [使用 CLI 來檢視日誌串流](#)
- [使用 重新建立失敗的叢集 rollback-on-failure](#)

在上檢視 CloudFormation 事件 CREATE_FAILED

您可以使用 主控台或 AWS ParallelCluster CLI 檢視CREATE_FAILED錯誤時的 CloudFormation 事件，以協助尋找根本原因。

主題

- [在 CloudFormation 主控台中檢視事件](#)
- [使用 CLI 在上檢視和篩選 CloudFormation 事件 CREATE_FAILED](#)

在 CloudFormation 主控台中檢視事件

若要查看導致"CREATE_FAILED"狀態的原因的詳細資訊，您可以使用 CloudFormation 主控台。

從主控台檢視 CloudFormation 錯誤訊息。

1. 登入 AWS 管理主控台 並導覽至 <https://console.aws.amazon.com/cloudformation>。
2. 選取名為 *cluster_name* 的堆疊。
3. 選擇事件索引標籤。
4. 透過按邏輯 ID 捲動資源事件清單，檢查無法建立之資源的狀態。如果子任務無法建立，請向後工作以尋找失敗的資源事件。
5. 例如，如果您看到下列狀態訊息，您必須使用不超過目前 vCPU 限制或請求更多 vCPU 容量的執行個體類型。

```
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 在上檢視和篩選 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 AWS Command Line Interface 《使用者指南》中的 [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"
  }
]
```

在先前的範例中，失敗是在前端節點設定中。

使用 CLI 來檢視日誌串流

若要偵錯這類問題，您可以 [pcluster list-cluster-log-streams](#) 篩選 `node-type` 然後分析日誌串流內容，以使用 列出前端節點可用的日誌串流。

```
$ 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日誌串流如下：

- `cfn-init` 是 `cfn-init` 指令碼的日誌。首先檢查此日誌串流。您可能會在此日誌中看到 `Command chef failed` 錯誤。查看此行前面的行，以取得與錯誤訊息連線的更多詳細資訊。如需詳細資訊，請參閱 [cfn-init](#)。
- `cloud-init` 是 [cloud-init](#) 的日誌。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 擷取日誌串流的內容 [pcluster get-cluster-log-events](#) (請注意限制擷取事件數量 `--limit 5` 的選項) :

```
$ 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"
}
]
}

```

在先前的範例中，失敗是由runpostinstall失敗所造成，因此與組態OnNodeConfigured參數中使用的自訂引導指令碼內容嚴格相關[CustomActions](#)。

使用 重新建立失敗的叢集 **rollback-on-failure**

AWS ParallelCluster 在日誌群組中建立叢集 CloudWatch 日誌串流。您可以在 CloudWatch 主控台自訂儀表板或日誌群組中檢視這些日誌。如需詳細資訊，請參閱與 [Amazon CloudWatch Logs 的整合](#)及[Amazon CloudWatch 儀表板](#)。如果沒有可用的日誌串流，則失敗可能是由[CustomActions](#)自訂引導指令碼或 AMI 相關問題所造成。若要診斷在此情況下的建立問題，請使用 [再次建立叢集pcluster create-cluster](#)，包括將 `--rollback-on-failure` 參數設定為 `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": "3.15.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
$ pcluster ssh --cluster-name mycluster

```

登入前端節點後，您應該會找到三個主要日誌檔案，可用來尋找錯誤。

- `/var/log/cfn-init.log` 是 `cfn-init` 指令碼的日誌。首先檢查此日誌。您可能會在此日誌 `Command chef failed` 中看到錯誤，例如。查看此行前面的行，以取得與錯誤訊息連線的更多詳細資訊。如需詳細資訊，請參閱 [cfn-init](#)。
- `/var/log/cloud-init.log` 是 [cloud-init](#) 的日誌。如果您在 中沒有看到任何內容 `cfn-init.log`，請嘗試接下來檢查此日誌。
- `/var/log/cloud-init-output.log` 是 [cloud-init](#) 所執行命令的輸出。這包括來自 的輸出 `cfn-init`。在大多數情況下，您不需要查看此日誌來疑難排解此類問題。

使用 Terraform 對叢集部署進行故障診斷

本節與使用 Terraform 部署的叢集相關。

找不到 ParallelCluster API

規劃可能會失敗，因為找不到 ParallelCluster API。在這種情況下，傳回的錯誤會如下：

```
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 Terraform 建立叢集”](#)。

使用者無權呼叫 ParallelCluster API

規劃可能會失敗，因為您擔任部署 Terraform 專案的 IAM 角色/使用者沒有與 ParallelCluster API 互動的許可。在這種情況下，傳回的錯誤會如下：

```
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**
}
```

對擴展問題進行故障診斷

本節與使用 3.0.0 版及更新 AWS ParallelCluster 版本搭配 Slurm 任務排程器安裝的叢集相關。如需設定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。這可避免產生任何非預期的成本。

```
$ pcluster update-compute-fleet --cluster-name mycluster \
  --status STOP_REQUESTED
```

您可以使用 [pcluster list-cluster-log-streams](#) 命令列出叢集節點可用的日誌串流，並使用其中一個失敗節點 `private-dns-name` 的 或前端節點進行篩選：

```
$ 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，您可以使用 [pcluster get-cluster-log-events](#) 命令擷取日誌串流的內容來分析日誌串流，並將 `--log-stream-name` 對應的 傳遞至下節提到的其中一個金鑰日誌：

```
$ 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 在日誌群組中建立叢集 CloudWatch 日誌串流。您可以在 CloudWatch 主控台自訂儀表板或日誌群組中檢視這些日誌。如需詳細資訊，請參閱[與 Amazon CloudWatch Logs 的整合](#)及[Amazon CloudWatch 儀表板](#)。

主題

- [除錯的金鑰日誌](#)
- [在我無法執行任務slurm_resume.log時看到InsufficientInstanceCapacity錯誤，或在我無法建立叢集clustermgtd.log時看到錯誤](#)
- [故障診斷節點初始化問題](#)
- [對非預期的節點替換和終止進行故障診斷](#)
- [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](#)
- [故障診斷其他已知節點和任務問題](#)

除錯的金鑰日誌

下表提供前端節點金鑰日誌的概觀：

- /var/log/cfn-init.log - 這是 CloudFormation 初始化日誌。它包含設定執行個體時執行的所有命令。使用它來疑難排解初始化問題。
- /var/log/chef-client.log - 這是 Chef 用戶端日誌。它包含透過 Chef/CINC 執行的所有命令。使用它來疑難排解初始化問題。
- /var/log/parallelcluster/slurm_resume.log - 這是ResumeProgram日誌。它會啟動動態節點的執行個體。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日誌。當動態節點的執行個體終止時，稱為。使用它來疑難排解動態節點終止問題。檢查此日誌時，您也應該檢查clustermgtd日誌。
- /var/log/parallelcluster/clustermgtd - 這是clustermgtd日誌。它執行為管理大多數叢集操作動作的集中式協助程式。使用它來疑難排解任何啟動、終止或叢集操作問題。
- /var/log/slurmctld.log - 這是Slurm控制常駐程式 log。AWS ParallelCluster doesn 不會做出擴展決策。相反地，它只會嘗試啟動資源以滿足Slurm需求。它適用於擴展和配置問題、任務相關問題，以及任何排程器相關的啟動和終止問題。

- `/var/log/parallelcluster/compute_console_output` - 此日誌記錄來自靜態運算節點範例子集的主控制台輸出，這些節點已意外終止。如果靜態運算節點終止且運算節點日誌無法在 CloudWatch 中使用，請使用此日誌。當您使用 Amazon EC2 主控台或 AWS CLI 擷取執行個體主控台輸出時，您收到 `compute_console_output log` 的內容相同。

以下是運算節點的金鑰日誌：

- `/var/log/cloud-init-output.log` - 這是 [cloud-init](#) 日誌。它包含設定執行個體時執行的所有命令。使用它來疑難排解初始化問題。
- `/var/log/parallelcluster/computemgtd` - 這是 `computemgtd` 日誌。它會在每個運算節點上執行，以監控頭節點上 `clustermgtd` 協助程式離線的不常見事件中的節點。使用它來疑難排解非預期的終止問題。
- `/var/log/slurmd.log` - 這是 Slurm 運算協助程式日誌。使用它來疑難排解初始化和運算失敗問題。

在我無法執行任務 `slurm_resume.log` 時看到 `InsufficientInstanceCapacity` 錯誤，或在我無法建立叢集 `clustermgtd.log` 時看到錯誤

如果叢集使用 Slurm 排程器，表示您遇到容量不足的問題。如果提出執行個體啟動請求時沒有足夠的執行個體可用，則會傳回 `InsufficientInstanceCapacity` 錯誤。

對於靜態執行個體容量，您可以在的 `clustermgtd` 日誌中找到錯誤 `/var/log/parallelcluster/clustermgtd`。

對於動態執行個體容量，您可以在的 `ResumeProgram` 日誌中找到錯誤 `/var/log/parallelcluster/slurm_resume.log`。

訊息看起來與下列範例類似：

```
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來避免取得這些類型的錯誤訊息：

- 如果啟用放置群組，請將其停用。如需詳細資訊，請參閱 [置放群組和執行個體啟動問題](#)。

- 預留執行個體的容量，並使用 ODCR（隨需容量預留）啟動它們。如需詳細資訊，請參閱[使用隨需容量預留 \(ODCR\) 啟動執行個體](#)。
- 使用不同的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，您可以使用多個運算資源來利用快速容量不足容錯移轉。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，並利用多個執行個體類型配置。如需設定多個執行個體的詳細資訊，請參閱 [使用 Slurm 進行多個執行個體類型配置](#)和 [Scheduling / SlurmQueues / ComputeResources / Instances](#)。
- 透過變更叢集組態 [Scheduling / SlurmQueues / Networking](#) 中的子網路 ID，將佇列移至不同的可用區域[SubnetIds](#)。
- 如果您的工作負載未緊密耦合，請將佇列跨越不同的可用區域。如需設定多個子網路的詳細資訊，請參閱 [Scheduling / SlurmQueues / Networking / SubnetIds](#)。

故障診斷節點初始化問題

本節說明如何對節點初始化問題進行疑難排解。這包括節點無法啟動、開啟電源或加入叢集的問題。

主題

- [前端節點](#)
- [運算節點](#)

前端節點

適用的日誌：

- /var/log/cfn-init.log
- /var/log/chef-client.log
- /var/log/parallelcluster/clustermgtd
- /var/log/parallelcluster/slurm_resume.log
- /var/log/slurmctld.log

檢查 /var/log/cfn-init.log和 /var/log/chef-client.log日誌或對應的日誌串流。這些日誌包含設定前端節點時執行的所有動作。在設定期間發生的大多數錯誤都應有位於 /

`/var/log/chef-client.log` 日誌中的錯誤訊息。如果在叢集的組態中指定 `OnNodeStart` 或 `OnNodeConfigured` 指令碼，請再次檢查指令碼是否透過日誌訊息成功執行。

建立叢集時，前端節點必須等待運算節點加入叢集，才能加入叢集。因此，如果運算節點無法加入叢集，則前端節點也會失敗。您可以根據您使用的運算備註類型，遵循下列其中一組程序，來疑難排解此類問題：

運算節點

- 適用的日誌：
 - `/var/log/cloud-init-output.log`
 - `/var/log/slurmd.log`
- 如果啟動運算節點，請先檢查 `/var/log/cloud-init-output.log`，其中應包含類似於前端節點上日誌的設定/`/var/log/chef-client.log`日誌。在設定期間發生的大多數錯誤都應在 `/var/log/cloud-init-output.log`日誌中顯示錯誤訊息。如果在叢集組態中指定預先安裝或安裝後指令碼，請檢查它們是否已成功執行。
- 如果您使用自訂 AMI 修改Slurm組態，則可能會發生 Slurm相關錯誤，導致運算節點無法加入叢集。如需排程器相關錯誤，請檢查`/var/log/slurmd.log`日誌。

動態運算節點：

- 搜尋您運算節點名稱的ResumeProgram日誌 (`/var/log/parallelcluster/slurm_resume.log`)，以查看是否ResumeProgram曾經使用節點呼叫。(如果ResumeProgram從未呼叫，您可以檢查`slurmctld`日誌 (`/var/log/slurmctld.log`) 來判斷是否Slurm曾經嘗試ResumeProgram使用節點呼叫)。
- 請注意，不正確的許可ResumeProgram可能會導致以無提示方式ResumeProgram失敗。如果您使用自訂 AMI 並修改ResumeProgram設定，請檢查 ResumeProgram 是否為slurm使用者所擁有，並具有 `744(rwxr--r--)` 許可。
- 如果呼叫 ResumeProgram，請檢查是否已為節點啟動執行個體。如果未啟動任何執行個體，您可以看到描述啟動失敗的錯誤訊息。
- 如果執行個體已啟動，則設定程序期間可能發生問題。您應該會從ResumeProgram日誌中看到對應的私有 IP 地址和執行個體 ID。此外，您可以查看特定執行個體的對應設定日誌。如需使用運算節點對設定錯誤進行故障診斷的詳細資訊，請參閱下一節。

靜態運算節點：

- 檢查 `clustermgtd(/var/log/parallelcluster/clustermgtd)` 日誌，查看是否已為節點啟動執行個體。如果未啟動，則應該會出現明確錯誤訊息，詳細說明啟動失敗。
- 如果執行個體已啟動，則設定程序期間會發生一些問題。您應該會從 `ResumeProgram` 日誌中看到對應的私有 IP 地址和執行個體 ID。此外，您可以查看特定執行個體對應的設定日誌。

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，且任務仍處於 PD（待定狀態），請再次檢查 `/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，`AWSServiceRoleForEC2Spot` 服務連結角色必須存在於您的帳戶中。若要使用在帳戶中建立此角色 AWS CLI，請執行下列命令：

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

如需詳細資訊，請參閱 AWS ParallelCluster 《Amazon EC2 使用者指南[使用 競價型執行個體](#)》中的 [Spot 執行個體請求的使用者指南和服務連結角色](#)。Amazon EC2

對非預期的節點替換和終止進行故障診斷

本節繼續探索如何對節點相關問題進行故障診斷，特別是當節點遭到意外取代或終止時。

- 適用的日誌：
 - `/var/log/parallelcluster/clustermgtd`（前端節點）
 - `/var/log/slurmctld.log`（前端節點）
 - `/var/log/parallelcluster/computemgtd`（運算節點）

節點意外取代或終止

- 檢查 `clustermgtd` 日誌 (`/var/log/parallelcluster/clustermgtd`) 中的 是否 `clustermgtd` 取代或終止節點。請注意，`clustermgtd` 會處理所有正常節點維護動作。
- 如果 `clustermgtd` 取代或終止節點，則應該會出現一則訊息，詳細說明為什麼對節點採取此動作。如果原因與排程器相關（例如，因為節點位於 DOWN），請查看 `slurmctld` 日誌以取得更多資訊。如果原因是與 Amazon EC2 相關，則應有資訊性訊息詳細說明需要替換的 Amazon EC2 相關問題。
- 如果 `clustermgtd` 未終止節點，請先檢查這是否為 Amazon EC2 的預期終止，特別是 Spot 終止。如果 `clustermgtd` 被判定為運作狀態不佳，`computemgtd` 則在運算節點上執行的 也可以終止節點。檢查 `computemgtd` 日誌 (`/var/log/parallelcluster/computemgtd`) 以查看節點是否已 `computemgtd` 終止。

節點失敗

- 簽入 `slurmctld` 日誌 (`/var/log/slurmctld.log`) 以查看任務或節點失敗的原因。請注意，如果節點失敗，任務會自動重新排入佇列。
- 如果 `slurm_resume` 報告啟動節點，並在幾分鐘後 `clustermgtd` 報告該節點的 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。若要從運算 (`/var/log/cloud-init-output.log`) 擷取日誌，請執行下列步驟：
 - 提交任務，讓 Slurm 啟動新的節點。
 - 等待運算節點啟動。
 - 修改執行個體啟動的關閉行為，以便停止失敗的運算節點，而不是終止。

```
$ 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
```

- 標記節點以方便識別。

```
$ aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=Name,Value=QUARANTINED-Compute
```

- 變更 `parallelcluster:cluster-name` 標籤，從叢集分離節點。

```
$ 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

- 適用的日誌：
 - /var/log/parallelcluster/clustermgtd (前端節點)
 - /var/log/parallelcluster/slurm_suspend.log (前端節點)
- 在大多數情況下，會clustermgtd處理所有預期的執行個體終止動作。請檢查clustermgtd日誌，了解為何無法取代或終止節點。
- 對於失敗的動態節點[SlurmSettings 屬性](#)，請檢查SuspendProgram日誌中的 是否SuspendProgram以特定節點做為引數slurmctld呼叫。請注意，實際上SuspendProgram不會執行任何動作。相反地，它只會在呼叫時記錄。所有執行個體終止和NodeAddr重設都是由完成clustermgtd。會在之後SuspendTimeout自動Slurm讓節點回到 POWER_SAVING 狀態。
- 如果運算節點因為引導失敗而持續失敗，請確認它們是否在[Slurm 叢集保護模式](#)啟用的情況下啟動。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。對引導指令碼進行故障診斷和修正。

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`，您的叢集可能 [Slurm 叢集保護模式](#) 已啟用，且佇列在預先定義的期間內已設定為 INACTIVE 狀態。

故障診斷其他已知節點和任務問題

另一種已知問題類型是 AWS ParallelCluster 可能無法配置任務或做出擴展決策。發生這類問題時，AWS ParallelCluster 只會根據 Slurm 指示啟動、終止或維護資源。針對這些問題，請檢查 `slurmctld` 日誌進行故障診斷。

置放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聯網骨幹。如果提出請求時沒有足夠的執行個體可用，則會傳回 `InsufficientInstanceCapacity` 錯誤。若要降低在使用叢集置放群組時收到此錯誤的可能性，請將 [SlurmQueues](#) / [Networking](#) / [PlacementGroup](#) / [Enabled](#) 參數設定為 `false`。

如需額外控制容量存取，請考慮 [使用 ODCR \(隨需容量預留 \) 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 [對執行個體啟動問題和置放群組角色和限制進行故障診斷](#)。 <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#concepts-placement-groups>

取代目錄

某些目錄無法取代。如果您在取代目錄時遇到問題，可能是這種情況。下列目錄在節點之間共用，無法取代。

- `/opt/intel` - 這包括 Intel MPI、Intel Parallel Studio 和相關檔案。
- `/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。)

對 Amazon DCV 中的問題進行故障診斷

主題

- [Amazon DCV 的日誌](#)
- [Ubuntu Amazon DCV 問題](#)

Amazon DCV 的日誌

Amazon DCV 的日誌會寫入 `/var/log/dcv/` 目錄中的檔案。檢閱這些日誌有助於對問題進行疑難排解。

執行個體類型應至少有 1.7 GB (GiB) 的 RAM 才能執行 Amazon DCV。Nano 和微型執行個體類型沒有足夠的記憶體來執行 Amazon DCV。

AWS ParallelCluster 在日誌群組中建立 Amazon DCV 日誌串流。您可以在 CloudWatch 主控台自訂儀表板或日誌群組中檢視這些日誌。如需詳細資訊，請參閱[與 Amazon CloudWatch Logs 的整合](#)及[Amazon CloudWatch 儀表板](#)。

Ubuntu Amazon DCV 問題

透過 Ubuntu 上的 Amazon DCV 工作階段執行 Gnome 終端機時，您可能無法自動存取透過登入 shell AWS ParallelCluster 提供的使用者環境。使用者環境提供 Openmpi 或 intelmpi 等環境模組，以及其他使用者設定。

Gnome 終端機的預設設定可防止 Shell 以登入 Shell 身分啟動。這表示 shell 設定檔不會自動取得，也不會載入 AWS ParallelCluster 使用者環境。

若要正確取得 shell 描述檔並存取 AWS ParallelCluster 使用者環境，請執行下列其中一項操作：

- 變更預設終端機設定：
 1. 在 Gnome 終端機中選擇編輯功能表。
 2. 選取偏好設定，然後選取設定檔。
 3. 選擇命令，然後選取執行命令做為登入 shell。
 4. 開啟新的終端機。
- 使用命令列來取得可用的設定檔：

```
$ source /etc/profile && source $HOME/.bashrc
```

使用 AWS Batch 整合對叢集中的問題進行故障診斷

本節為具有 AWS Batch 排程器整合的叢集提供可能的疑難排解秘訣，特別是前端節點問題、運算問題、任務失敗和逾時錯誤。

主題

- [前端節點問題](#)
- [運算問題](#)
- [任務失敗](#)
- [端點 URL 錯誤的連線逾時](#)

前端節點問題

您可以使用與Slurm叢集相同的方式對前端節點設定問題進行故障診斷 (Slurm特定日誌除外)。如需有關這些問題的詳細資訊，請參閱[前端節點](#)。

運算問題

AWS Batch 會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](#)文件以取得協助。

任務失敗

如果任務失敗，您可以執行 [awsbout](#) 命令來擷取任務輸出。您也可以執行 [awsbstat](#) 命令，以取得 Amazon CloudWatch 所存放任務日誌的連結。

端點 URL 錯誤的連線逾時

如果多節點平行任務失敗並發生錯誤：Connect timeout on endpoint URL

- 在awsbout輸出日誌中，檢查任務是否與輸出平行多節點：Detected 3/3 compute nodes. Waiting for all compute nodes to start.
- 驗證運算節點子網路是否為公有。

AWS Batch 使用時，多節點平行任務不支援使用公有子網路 AWS ParallelCluster。為您的運算節點和任務使用私有子網路。如需詳細資訊，請參閱AWS Batch 《使用者指南》中的[運算環境考量](#)。若要為您的運算節點設定私有子網路，請參閱 [AWS ParallelCluster 使用 AWS Batch 排程器](#)。

對與 Active Directory 的多使用者整合進行故障診斷

本節與與 Active Directory 整合的叢集相關。

如果 Active Directory 整合功能未如預期般運作，SSSD 日誌可以提供有用的診斷資訊。這些日誌位於叢集節點/var/log/sss的 中。根據預設，它們也會存放在叢集的 Amazon CloudWatch 日誌群組中。

主題

- [Active Directory 特定疑難排解](#)
- [啟用除錯模式](#)
- [如何從 LDAPS 移至 LDAP](#)

- [如何停用 LDAPS 伺服器憑證驗證](#)
- [如何使用 SSH 金鑰而非密碼登入](#)
- [如何重設使用者密碼和過期的密碼](#)
- [如何驗證加入的網域](#)
- [如何疑難排解憑證的問題](#)
- [如何驗證與 Active Directory 的整合是否正常運作](#)
- [如何對登入運算節點進行故障診斷](#)
- [多使用者環境中 SimCenter StarCCM+ 任務的已知問題](#)
- [使用者名稱解析的已知問題](#)
- [如何解決主目錄建立問題](#)

Active Directory 特定疑難排解

本節與針對 Active Directory 類型進行疑難排解相關。

Simple AD

- 此DomainReadOnlyUser值必須符合使用者的 Simple AD 目錄基礎搜尋：

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

cn 的注意事項Users。

- 預設管理員使用者為 Administrator。
- Ldapsearch 在使用者名稱之前需要 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

- DomainReadOnlyUser 值必須符合 使用者的 AWS Managed Microsoft AD 目錄基礎搜尋：

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


- 預設管理員使用者為 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

從 SSSD 偵錯日誌有助於對問題進行疑難排解。若要啟用偵錯模式，您必須使用對叢集組態所做的下列變更來更新叢集：

```
DirectoryService:  
  AdditionalSssdConfigs:  
    debug_level: "0x1fff"
```

如何從 LDAPS 移至 LDAP

不建議從 LDAPS (LDAP 搭配 TLS/SSL) 移至 LDAP，因為僅 LDAP 不提供任何加密。不過，它對於測試目的和疑難排解很有用。

您可以使用先前的組態定義更新叢集，將叢集還原至先前的組態。

若要從 LDAPS 移至 LDAP，您必須在叢集組態中使用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

在前端節點上暫時停用 LDAPS 伺服器憑證驗證非常有用，可用於測試或故障診斷。

您可以使用先前的組態定義更新叢集，將叢集還原至先前的組態。

若要停用 LDAPS 伺服器憑證驗證，您必須在叢集組態中使用下列變更來更新叢集：

```
DirectoryService:  
  LdapTlsReqCert: never
```

如何使用 SSH 金鑰而非密碼登入

在您第一次使用密碼登入 `/home/$user/.ssh/id_rsa` 後，會在 `~/.ssh` 中建立 SSH 金鑰。若要使用 SSH 金鑰登入，您必須使用密碼登入，在本機複製 SSH 金鑰，然後照常將其用於 SSH 無密碼：

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY $username@$head_node_ip
```

如何重設使用者密碼和過期的密碼

如果使用者無法存取叢集，其 [AWS Managed Microsoft AD 密碼可能已過期](#)。

若要重設密碼，請對具有目錄寫入許可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. 更新叢集的新秘密值：
 - a. 使用 `pcluster update-compute-fleet` 命令停止運算機群。
 - b. 從叢集前端節點內執行下列命令。

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

密碼重設和叢集更新後，應還原使用者的叢集存取權。

如需詳細資訊，請參閱《[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，而非前端節點。

```
$ 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
```

如何疑難排解憑證的問題

當 LDAPS 通訊無法運作時，可能是由於 TLS 通訊中的錯誤，進而可能是憑證問題所致。

憑證的注意事項：

- 叢集組態中指定的憑證 `LdapTlsCaCert` 必須是一組 PEM 憑證，其中包含為網域控制站發行憑證的整個授權憑證 (CA) 鏈的憑證。
- PEM 憑證套件是由 PEM 憑證串連而成的檔案。
- PEM 格式的憑證 (通常用於 Linux) 等同於 base64 DER 格式的憑證 (通常由 Windows 匯出)。
- 如果網域控制站的憑證是由次級 CA 發行，則憑證套件必須包含次級 CA 和根 CA 的憑證。

對驗證步驟進行故障診斷：

下列驗證步驟假設命令是從叢集前端節點內執行，且網域控制器可在連線 `SERVER:PORT`。

若要對與憑證相關的問題進行疑難排解，請遵循下列驗證步驟：

驗證步驟：

1. 檢查與 Active Directory 網域控制站的連線：

確認您可以連線至網域控制器。如果此步驟成功，則與網域控制站的 SSL 連線成功，且憑證已驗證。您的問題與憑證無關。

如果此步驟失敗，請繼續下一個驗證。

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

2. 檢查憑證驗證：

確認本機 CA 憑證套件可以驗證網域控制器提供的憑證。如果此步驟成功，則您的問題與憑證無關，但與其他聯網問題有關。

如果此步驟失敗，請繼續下一個驗證。

```
$ openssl verify -verbose -  
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

3. 檢查 Active Directory 網域控制站提供的憑證：

驗證網域控制站提供的憑證內容是否如預期。如果此步驟成功，您可能會遇到用於驗證控制器的 CA 憑證問題，請前往下一個故障診斷步驟。

如果此步驟失敗，您必須更正為網域控制站發行的憑證，並重新執行故障診斷步驟。

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

4. 檢查憑證的內容：

驗證網域控制站提供的憑證內容是否如預期。如果此步驟成功，您可能會遇到用於驗證控制器的 CA 憑證問題，請前往下一個疑難排解步驟。

如果此步驟失敗，您必須更正為網域控制站發行的憑證，並重新執行故障診斷步驟。

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

5. 檢查本機 CA 憑證套件的內容：

驗證用於驗證網域控制站憑證的本機 CA 憑證套件內容是否如預期。如果此步驟成功，您可能會遇到網域控制站提供的憑證問題。

如果此步驟失敗，您必須更正為網域控制站發行的 CA 憑證套件，並重新執行故障診斷步驟。

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

如何驗證與 Active Directory 的整合是否正常運作

如果以下兩個檢查成功，則與 Active Directory 的整合正在運作。

檢查：

1. 您可以探索目錄中定義的使用者：

從叢集前端節點內，做為 ec2-user：

```
$ getent passwd $ANY_AD_USER
```

2. 您可以在提供使用者密碼的前端節點中 SSH：

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

如果檢查一失敗，我們預期檢查二也會失敗。

其他疑難排解檢查：

- 確認使用者存在於目錄中。
- 啟用[偵錯記錄](#)。
- 考慮從[LDAPS 移至 LDAP](#) 暫時停用加密，以排除 LDAPS 問題。

如何對登入運算節點進行故障診斷

本節與登入以運算與 Active Directory 整合之叢集中的節點有關。

使用時 AWS ParallelCluster，設計會停用叢集運算節點的密碼登入。

所有使用者都必須使用自己的 SSH 金鑰登入運算節點。

如果在叢集組態中[GenerateSshKeysForUsers](#)啟用，使用者可以在第一次身分驗證（例如登入）之後，在前端節點中擷取其 SSH 金鑰。

當使用者第一次在前端節點上進行身分驗證時，他們可以擷取以目錄使用者身分自動為其產生的 SSH 金鑰。也會為使用者建立主目錄。當 sudo 使用者第一次切換到前端節點中的使用者時，也可能發生這種情況。

如果使用者尚未登入前端節點，則不會產生 SSH 金鑰，而且使用者將無法登入運算節點。

多使用者環境中 SimCenter StarCCM+ 任務的已知問題

本節與 Simcenter StarCCM+ 運算流體動力學軟體從 Siemens 在多使用者環境中啟動的任務相關。

如果您執行設定為使用內嵌 IntelMPI 的 StarCCM+ v16 任務，預設會使用 SSH 引導 MPI 程序。

由於已知的[Slurm錯誤](#)導致使用者名稱解析錯誤，任務可能會失敗並出現等錯誤error setting up the bootstrap proxies。此錯誤只會影響 3.1.1 和 3.1.2 AWS ParallelCluster 版。

若要防止這種情況發生，請強制 IntelMPI 使用 Slurm做為 MPI 引導方法。將環境變數匯出I_MPI_HYDRA_BOOTSTRAP=slurm至啟動 StarCCM+ 的任務指令碼，如 [IntelMPI 官方文件](#)所述。

使用者名稱解析的已知問題

本節與在任務中擷取使用者名稱相關。

由於 [Slurm 中的已知錯誤](#)，nobody如果您在沒有 的情況下執行任務，則任務程序中擷取的使用者名稱可能是 srun。此錯誤只會影響 3.1.1 和 3.1.2 AWS ParallelCluster 版。

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，nobody可能會傳回為使用者名稱。

您可以使用下列解決方法。

1. 如果可能 'sbatch'，請使用 'srun' 而非 啟動您的任務。
2. 在叢集組態中設定 [AdditionalSssdConfigs](#) 以啟用 SSSD 列舉，如下所示。

```
AdditionalSssdConfigs:  
  enumerate: true
```

如何解決主目錄建立問題

本節與主目錄建立問題相關。

如果您看到如下範例所示的錯誤，當您第一次登入前端節點時，不會為您建立主目錄。或者，當您第一次從 sudoer 切換到 head 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
```

主目錄建立失敗的原因可能是叢集前端節點中安裝的 `oddjob` 和 `oddjob-mkhomedir` 套件。

如果沒有主目錄和 SSH 金鑰，使用者就無法將任務或 SSH 提交至叢集節點。

如果您需要系統中的 `oddjob` 套件，請確認 `oddjobd` 服務正在執行中，並重新整理 PAM 組態檔案，以確保已建立主目錄。若要這樣做，請在前端節點中執行命令，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，請將其解除安裝並重新整理 PAM 組態檔案，以確保已建立主目錄。若要這樣做，請在前端節點中執行命令，如下列範例所示。

```
sudo yum remove -y oddjob oddjob-mkhomedir  
sudo authconfig --enablemkhomedir --updateall
```

對自訂 AMI 問題進行故障診斷

本節提供自訂 AMI 問題的可能疑難排解秘訣。

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，您可以忽略這些警告。

如果您未來不想看到這些警告，請使用下列標籤來標記自訂 AMI，其中 *my-os* 是 alinux2、alinux2023、rhel8、ubuntu2404、ubuntu2204 或 rhel9 *"3.15.0"* 使用的 pcluster 版本之一：

```
$ aws ec2 create-tags \
  --resources ami-yourcustomAmi \
  --tags Key="parallelcluster:version",Value="3.15.0"
Key="parallelcluster:os",Value="my-os"
```

在 cfn-hup 未執行時對叢集更新逾時進行故障診斷

cfn-hup 協助程式是一個常駐程式，它會偵測資源中繼資料中的變更，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 協助程式是由 啟動 supervisor。但在啟動後，cfn-hup 程序會與 supervisor 控制項分離。如果 cfn-hup 魔抗被外部演員殺死，則不會自動重新啟動。如果 cfn-hup 未執行，在叢集更新期間，CloudFormation 堆疊會如預期啟動更新程序，但前端節點上不會啟動更新程序，且堆疊最終會逾時。從叢集日誌 /var/log/chef-client 中，您可以看到永遠不會叫用更新配方。

檢查並在失敗 cfn-hup 時重新啟動

1. 在前端節點上，檢查 是否 cfn-hup 正在執行：

```
$ ps aux | grep cfn-hup
```

2. 檢查前端節點 /var/log/supervisord.log 上的 cfn-hup 日誌 /var/log/cfn-hup.log 和。
3. 如果 cfn-hup 未執行，請嘗試執行以重新啟動它：


```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl
start cfn-hup
```

網路故障診斷

本節提供當您遇到網路問題時的疑難排解秘訣，特別是在單一公有子網路問題中處理叢集時。

單一公有子網路問題的叢集

cloud-init-output.log 從其中一個運算節點檢查。如果您發現類似以下內容，表示節點卡在 Slurm 初始化中，很可能是因為缺少 DynamoDB VPC 端點。新增 DynamoDB 端點。如需更多資訊，請參閱 [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#)。

```
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 自訂動作上的叢集更新失敗

當 [HeadNode // CustomActions OnNodeUpdated](#) 指令碼失敗時，更新會失敗，而且指令碼不會在復原時間執行。復原完成後，您有責任手動執行所需的清除。例如，如果 OnNodeUpdated 指令碼變更組態檔案中欄位的狀態（例如，從 true 到 false），然後失敗，您需要手動將該欄位值還原到更新前狀態（例如，false 到 true）。如需詳細資訊，請參閱 [自訂引導操作](#)。

使用自訂 Slurm 組態查看錯誤

從 3.6.0 AWS ParallelCluster 版開始，您無法再將單一 prolog 或 epilog 指令碼納入自訂 Slurm 組態中，以將其設為目標。在 3 AWS ParallelCluster .6.0 版和更新版本中，您必須在個別 prolog 和 Epilog 資料夾中找到自訂 Prolog 和 epilog 指令碼。這些資料夾預設為指向：

- 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 prolog 指令碼和 90_pcluster_noop epilog 指令碼。

Slurm 會以反向字母順序執行指令碼。Prolog 和 Epilog 資料夾都必須包含至少一個檔案。如需詳細資訊，請參閱[Slurm 與 prologepilog](#)及[Slurm 組態自訂](#)。

叢集警示

叢集運作狀態監控對於確保最佳效能至關重要。AWS ParallelCluster 可讓您監控叢集前端節點的多個 CloudWatch 型警示。

本節提供每種頭部節點叢集警示類型的詳細資訊，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：會發出頭部節點的整體狀態訊號。如果以下至少一個警示是，則為紅色。
- CLUSTER_NAME-HeadNode-Health：如果至少有一個 Amazon EC2 運作狀態檢查失敗，則為紅色。如果發生警示，建議您查看[故障診斷狀態檢查失敗的執行個體](#)。
- CLUSTER_NAME-HeadNode-Cpu：如果 CPU 使用率大於 90%，則為紅色。如果發生警示，請使用檢查消耗 CPU 最多的程序 `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。如果發生警示，請使用檢查最常耗用記憶體的程序 `ps -aux --sort=-%mem | head -n 10`。
- CLUSTER_NAME-HeadNode-Disk：如果路徑 / 上的佔用磁碟空間大於 90%，則為紅色。如果發生警示，請檢查使用佔用大部分空間的資料夾 `du -h --max-depth=2 / 2> /dev/null | sort -hr`。

解決導致錯誤或失敗的作業系統組態變更

對 AWS ParallelCluster 節點進行作業系統組態變更時，可能會發生各種可能導致叢集建立、更新或操作失敗的問題。本節提供識別和解決常見作業系統組態相關問題的指引。

常見的作業系統組態問題

地區設定組態問題

最常見的作業系統組態問題之一與地區設定有關。如果您看到如下錯誤：

```
cannot change locale (en_US.utf-8) because it has an invalid name
```

這通常發生在以下情況：

- yum 安裝程序失敗，並使地區設定處於不一致狀態
- 使用者提早終止安裝程序
- Locale 套件遺失或損毀

如何診斷

1. 檢查您是否可以切換到 pcluster-admin 使用者：

```
$ su - pcluster-admin
```

如果您看到類似的錯誤 `cannot change locale...no such file or directory`，這會確認問題。

2. 檢查可用的地區設定：

```
$ localedef --list
```

如果這傳回空清單或不包含預設地區設定，則您的地區設定組態會中斷。

3. 檢查最後一個 yum 命令：

```
$ yum history  
$ yum history info #ID
```

如果最後一個 ID 沒有 `Return-Code: Success`，安裝後指令碼可能無法成功執行。

如何解決

透過重新安裝語言套件來重建地區設定：

```
$ sudo yum reinstall glibc-all-langpacks
```

重建之後，請執行下列動作來確認問題已修正：

```
$ su - pcluster-admin
```

如果沒有出現錯誤或警告，表示問題已解決。

作業系統套件衝突

安裝自訂套件或修改系統套件時，可能會發生衝突，導致叢集無法正常運作。

如何診斷

1. 檢查 chef-client 日誌是否有套件相關的錯誤：

```
$ less /var/log/chef-client.log
```

2. 在 cfn-init 日誌中尋找套件相依性衝突：

```
$ less /var/log/cfn-init.log
```

如何解決

1. 如果特定套件造成問題，請嘗試重新安裝它：

```
$ sudo yum reinstall package-name
```

2. 對於相依性衝突，您可能需要移除衝突的套件：

```
$ sudo yum remove conflicting-package
```

3. 如果問題仍然存在，請考慮使用 `pcluster build-image` 命令預先安裝的必要套件來建立自訂 AMI。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。

系統組態檔案修改

修改重要的系統組態檔案可能會導致叢集失敗，特別是這些檔案由管理時 AWS ParallelCluster。

如何診斷

1. 檢查 Chef-client 日誌中提及特定組態檔案的錯誤：

```
$ grep -i "config" /var/log/chef-client.log
```

2. 尋找組態檔案中的許可或語法錯誤：

```
$ less /var/log/cfn-init.log
```

如何解決

1. 將修改後的組態檔案還原為其原始狀態：

```
$ sudo cp /etc/file.conf.bak /etc/file.conf
```

2. 如果您需要持續變更系統組態檔案，請使用自訂引導操作，而不是直接修改檔案：

```
HeadNode:
  CustomActions:
    OnNodeConfigured:
      Script: s3://bucket-name/config-script.sh
```

如需詳細資訊，請參閱 [自訂引導操作](#)。

3. 對於必須直接對系統檔案進行的組態變更，請考慮建立自訂 AMI。如需詳細資訊，請參閱 [AWS ParallelCluster AMI 自訂](#)。

核心更新和相容性問題

核心更新可能會導致特定 AWS 服務的相容性問題，特別是 Amazon FSx for Lustre。

如何診斷

1. 檢查是否已套用核心更新：

```
$ uname -r
```

2. 在日誌中尋找 Amazon FSx 掛載失敗：

```
$ grep -i "fsx" /var/log/chef-client.log
```

如何解決

1. 對於 Ubuntu 22.04，請避免更新到最新的核心，因為該核心沒有 Amazon FSx 用戶端。如需詳細資訊，請參閱[作業系統考量事項](#)。
2. 如果您已更新核心並遇到問題，請考慮降級至相容的核心版本：

```
$ sudo apt install linux-image-previous-version
```

3. 對於持久性核心自訂，請使用您需要的特定核心版本建立自訂 AMI。如需詳細資訊，請參閱[AWS ParallelCluster AMI 自訂](#)。

作業系統組態變更的最佳實務

若要將進行作業系統組態變更時的問題降至最低：

1. 使用自訂引導操作：使用 `OnNodeStart` 或 `OnNodeConfigured` 指令碼以受控制的方式進行變更，而不是直接修改系統檔案。如需詳細資訊，請參閱[自訂引導操作](#)。
2. 建立自訂 AMIs：對於重大作業系統修改，請使用 建立自訂 AMI，`pcluster build-image` 而不是變更執行中的執行個體。如需詳細資訊，請參閱[AWS ParallelCluster AMI 自訂](#)。
3. 測試變更優先：在將變更套用至生產叢集之前，請先在小型測試叢集上測試變更，以確保相容性。
4. 文件變更：追蹤所有作業系統組態變更，以利故障診斷。
5. 備份組態檔案：修改任何系統組態檔案之前，請先建立備份：

```
$ sudo cp /etc/file.conf /etc/file.conf.bak
```

6. 變更後檢查日誌：變更作業系統組態後，請檢查日誌是否有任何錯誤：

```
$ less /var/log/cfn-init.log  
$ less /var/log/chef-client.log
```

透過遵循這些準則，您可以將導致叢集故障的作業系統組態變更風險降至最低，並更有效地對發生的任何問題進行故障診斷。

AWS ParallelCluster 支援政策

AWS ParallelCluster 同時支援多個版本。每個 AWS ParallelCluster 版本都有排定的支援生命週期結束 (EOSL) 日期。在 EOSL 日期之後，不會為該版本提供進一步的支援或維護。

AWS ParallelCluster 使用 `major.minor.patch` 版本方案。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版本的新次要版本中。次要版本在主要版本中向後相容。對於重大問題，透過修補程式版本 AWS 提供修正，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版本的更新，您需要升級至新的次要或修補程式版本。

AWS ParallelCluster 版本	支援的生命週期結束 (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	2/28/2025
3.8.x	6/30/2025
3.9.x	09/05/2025
3.10.x	12/27/2025
3.11.x	03/25/2026
3.12.x	06/30/2026
3.13.x	09/30/2026

AWS ParallelCluster 版本	支援的生命週期結束 (EOSL) 日期
3.14.x	03/29/2027

中的安全性 AWS ParallelCluster

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，該架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間共同責任。[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 Cloud 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。在[AWS 合規計畫](#)中，第三方稽核人員會定期測試和驗證我們的安全有效性。若要了解適用的合規計畫 AWS ParallelCluster，請參閱[AWS 合規計畫的服務範圍合規](#)。
- 雲端的安全性 – 您的責任取決於您使用的特定 AWS 服務或服務。您也必須負責幾個其他相關因素，包括資料的敏感度、公司的要求，以及適用的法律和法規。

本文件說明如何在使用時套用共同責任模型 AWS ParallelCluster。下列主題說明如何設定 AWS ParallelCluster 以符合您的安全與合規目標。您也會了解如何 AWS ParallelCluster 以協助您監控和保護 AWS 資源的方式使用。

主題

- [所使用服務的安全性資訊 AWS ParallelCluster](#)
- [中的資料保護 AWS ParallelCluster](#)
- [的 Identity and Access Management AWS ParallelCluster](#)
- [的合規驗證 AWS ParallelCluster](#)
- [強制執行最低版本為 TLS 1.2](#)
- [設定受限環境的安全群組](#)

所使用服務的安全性資訊 AWS ParallelCluster

- [Amazon EC2 的安全性](#)
- [Amazon API Gateway 的安全性](#)
- [中的安全性 AWS Batch](#)
- [CloudFormation 中的安全性](#)
- [Amazon CloudWatch 的安全性](#)
- [AWS CodeBuild 中的安全性](#)

- [Amazon DynamoDB 的安全性](#)
- [Amazon ECR 的安全性](#)
- [Amazon ECS 的安全性](#)
- [Amazon EFS 的安全性](#)
- [FSx for Lustre 中的安全性](#)
- [AWS Identity and Access Management \(IAM\) 中的安全性](#)
- [EC2 Image Builder 的安全性](#)
- [中的安全性 AWS Lambda](#)
- [Amazon Route 53 的安全性](#)
- [Amazon SNS 的安全性](#)
- [Amazon SQS 的安全性 \(適用於 2.x AWS ParallelCluster 版\)。](#)
- [Amazon S3 的安全性](#)
- [Amazon VPC 的安全性](#)

中的資料保護 AWS ParallelCluster

AWS [共同責任模型](#)適用於 中的資料保護。如此模型所述，AWS 負責保護執行所有的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱[資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱AWS 安全性部落格上的[AWS 共同責任模型](#)和 [GDPR](#) 部落格文章。

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

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

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

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 AWS 使用者與服務互動所需的登入資料之外，本身不會存放任何客戶資料。

對於叢集中節點上的資料，可以靜態加密資料。

對於 Amazon EBS 磁碟區，加密是使用 [EbsSettings](#) 區段中的 [EbsSettings/Encrypted](#) 和 [EbsSettings/KmsKeyId](#) 設定進行設定。如需詳細資訊，請參閱《[Amazon EC2 使用者指南](#)》中的 [Amazon EBS 加密](#)。Amazon EC2

對於 Amazon EFS 磁碟區，使用 [EfsSettings](#) 區段中的 [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 [靜態加密的運作方式](#)。Amazon Elastic File System

對於 FSx for Lustre 檔案系統，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 FSx for Lustre 使用者指南](#)》中的 [加密靜態資料](#)。

對於具有 NVMe 磁碟區的執行個體類型，NVMe 執行個體存放磁碟區上的資料會使用在執行個體上的硬體模組上實作的 XTS-AES-256 密碼進行加密。加密金鑰是以硬體模組來產生，且對每個 NVMe 執行個體儲存體設備而言是唯一的。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 叫用將客戶資料傳輸至本機電腦以供儲存 AWS 的服務，請參閱該服務使用者指南中的安全與合規章節，以取得如何存放、保護和加密該資料的資訊。

傳輸中加密

根據預設，從執行 AWS ParallelCluster AWS 和服務端點的用戶端電腦傳輸的所有資料都會透過 HTTPS/TLS 連線傳送所有內容來加密。根據選取的執行個體類型，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 EC2 中的資料保護](#)
- [EC2 Image Builder 中的資料保護](#)
- [中的資料保護 CloudFormation](#)
- [Amazon EFS 中的資料保護](#)
- [Amazon S3 中的資料保護](#)
- [FSx for Lustre 中的資料保護](#)

的 Identity and Access Management AWS ParallelCluster

AWS ParallelCluster 使用 角色來存取您的 AWS 資源及其服務。AWS ParallelCluster 用於授予許可的執行個體和使用者政策記錄在 [AWS Identity and Access Management 中的 許可 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，但該使用者需要存取金鑰對，才能使用 執行相同的操作 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。

使用的登入資料 AWS ParallelCluster 會存放在純文字檔案中，不會加密。

- `$HOME/.aws/credentials` 檔案會存放存取 AWS 資源所需的長期登入資料。這些包含您的存取金鑰 ID 和私密存取金鑰。
- 短期憑證，像是您擔任的角色，或用於 AWS IAM Identity Center 服務的角色，也會分別儲存於 `$HOME/.aws/cli/cache` 和 `$HOME/.aws/sso/cache` 資料夾中。

降低風險

- 強烈建議您設定 `$HOME/.aws` 資料夾及其子資料夾和檔案的檔案系統權限，以限制為僅供授權使用者進行存取。
- 盡可能使用具有臨時憑證的角色，以在憑證洩漏時減少損害的機會。僅將長期憑證用於請求及重新整理短期角色憑證。

的合規驗證 AWS ParallelCluster

在多個合規計畫中，第三方稽核人員會評估 AWS 服務的安全性和 AWS 合規性。使用 AWS ParallelCluster 存取服務並不會改變該服務的合規性。

如需特定合規計畫範圍內 AWS 的服務清單，請參閱[AWS 合規計畫的合規計畫範圍內的服務](#)。如需一般資訊，請參閱[AWS 合規計畫](#)。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載 AWS Artifact 中的報告](#)。

使用時的合規責任 AWS ParallelCluster 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- [安全與合規快速入門指南](#) – 這些部署指南討論架構考量，並提供部署以安全與合規為中心之基準環境的步驟 AWS。
- [Amazon Web Services 上的 HIPAA 安全與合規架構 AWS 白皮書](#) – 此白皮書說明公司如何使用 AWS 來建立符合 HIPAA 規範的應用程式。
- [AWS 合規資源](#) – 此工作手冊和指南集合可能適用於您的產業和位置。
- 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符合內部實務、產業準則和法規的程度。
- [AWS Security Hub CSPM](#) – AWS 此服務提供 內安全狀態的完整檢視 AWS ，協助您檢查是否符合安全產業標準和最佳實務。

強制執行最低版本為 TLS 1.2

若要在與服務通訊時增加安全性 AWS ，您應該將 AWS ParallelCluster 設定為使用 TLS 1.2 或更新版本。當您使用時 AWS ParallelCluster ，Python 會用來設定 TLS 版本。

為了確保 AWS ParallelCluster 不使用早於 TLS 1.2 的 TLS 版本，您可能需要重新編譯 OpenSSL 以強制執行此最小值，然後重新編譯 Python 以使用新建置的 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 boto3
```

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 不會交涉早於 TLS 1.2 的任何內容，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。

設定受限環境的安全群組

根據預設，會 AWS ParallelCluster 建立和設定安全群組，以允許叢集節點之間的所有流量。在高度受限的環境中，您可能需要將網路存取限制為僅叢集操作所需的連接埠。本節說明如何為 AWS ParallelCluster 部署設定具有限制存取的自訂安全群組。

安全群組概觀

AWS ParallelCluster 使用安全群組來控制前端節點、運算節點和登入節點之間的網路流量 (如果已設定)。根據預設，當 AWS ParallelCluster 建立叢集時，它會建立安全群組，以允許叢集內節點之間的所有流量。在具有嚴格安全要求的環境中，您可以提供自訂安全群組，將流量限制為僅必要的連接埠。

您可以在叢集組態的下列區段中設定安全群組：

- [HeadNode / Networking](#) - 控制對前端節點的存取
- [Scheduling / SlurmQueues / Networking](#) - 控制對運算節點的存取
- [LoginNodes](#) - 控制登入節點的存取 (如果已設定)

對於這些區段，您可以指定：

- `SecurityGroups` - 取代 AWS ParallelCluster 將建立的預設安全群組
- `AdditionalSecurityGroups` - 新增安全群組，以及 建立的預設安全群組 AWS ParallelCluster

叢集操作所需的連接埠

設定自訂安全群組時，您必須確保在適當的節點之間開啟下列連接埠：

前端節點所需的連接埠

連線埠	通訊協定	Direction	用途
22	TCP	傳入	前端節點的 SSH 存取 (從允許的 IP 範圍)
6817-6819	TCP	傳入	Slurm 控制器連接埠 (來自運算和登入節點)
6817-6819	TCP	傳出	Slurm 控制器連接埠 (用於運算和登入節點)
8443	TCP	傳入	NICE DCV (如果啟用, 從允許的 IP 範圍)
2049 年 111 月	TCP/UDP	傳入	NFS (使用 NFS 進行共用儲存時, 來自運算和登入節點)
443	TCP	傳出	HTTPS 存取 AWS 服務 (如果未使用 VPC 端點)

運算節點所需的連接埠

連線埠	通訊協定	Direction	用途
22	TCP	傳入	SSH 存取 (來自前端節點和登入節點)
6818	TCP	傳入	Slurm 協助程式連接埠 (來自前端節點)
6817-6819	TCP	傳出	Slurm 控制器連接埠 (前端節點)
2049 年 111 月	TCP/UDP	傳出	NFS (對於前端節點, 如果使用 NFS 進行共用儲存)
443	TCP	傳出	HTTPS 存取 AWS 服務 (如果未使用 VPC 端點)

如果您使用的是 EFA（彈性布料轉接器），您還必須允許已啟用 EFA 的運算節點之間的所有流量：

- 運算節點與 EFA 之間的所有 TCP 和 UDP 流量
- 具有 EFA 的運算節點之間的 EFA 裝置上所有流量

Note

如果您使用的是 FSx for Lustre、Amazon EFS 或其他儲存解決方案等共用儲存系統，您也需要確保為這些服務開啟適當的連接埠。

建立自訂安全群組

若要為您的 AWS ParallelCluster 部署建立自訂安全群組，請遵循下列步驟：

1. 使用 AWS 管理主控台、AWS CLI 或 AWS CloudFormation 為前端節點、運算節點和登入節點（如適用）建立安全群組。
2. 設定安全群組規則以僅允許必要的流量，如上一節所述。
3. 參考叢集組態檔案中的這些安全群組。

以下是如何使用 CLI AWS 建立安全群組的範例：

```
# Create security group for head node
aws ec2 create-security-group \
  --group-name pcluster-head-node-sg \
  --description "Security group for ParallelCluster head node" \
  --vpc-id vpc-12345678

# Create security group for compute nodes
aws ec2 create-security-group \
  --group-name pcluster-compute-node-sg \
  --description "Security group for ParallelCluster compute nodes" \
  --vpc-id vpc-12345678

# Add rules to allow necessary traffic between head and compute nodes
# (Add specific rules based on the required ports listed above)
```

在叢集組態中設定安全群組

建立自訂安全群組之後，您可以在叢集組態檔案中參考這些群組：

```
# Example cluster configuration with custom security groups
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678
    SecurityGroups:
      - sg-headnode12345 # Custom security group for head node
  # Or use AdditionalSecurityGroups if you want to keep the default security groups
  # AdditionalSecurityGroups:
  #   - sg-additional12345
  ...

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
    Networking:
      SubnetIds:
        - subnet-12345678
      SecurityGroups:
        - sg-computenode12345 # Custom security group for compute nodes
      # Or use AdditionalSecurityGroups if you want to keep the default security
      groups
      # AdditionalSecurityGroups:
      #   - sg-additional12345
    ...

# If using login nodes
LoginNodes:
  Pools:
    - Name: login-pool
    ...
    Networking:
      SubnetIds:
        - subnet-12345678
      SecurityGroups:
        - sg-loginnode12345 # Custom security group for login nodes
```

```
# Or use AdditionalSecurityGroups if you want to keep the default security
groups
# AdditionalSecurityGroups:
#   - sg-additional12345
...
```

使用時 `SecurityGroups`，只會 AWS ParallelCluster 使用您指定的安全群組，取代預設的安全群組。使用時 `AdditionalSecurityGroups`，AWS ParallelCluster 會使用其建立的預設安全群組，以及您指定的其他安全群組。

Warning

如果您為運算執行個體啟用 [Elastic Fabric Adapter \(EFA\)](#)，請確定已啟用 EFA 的執行個體是允許所有傳入和傳出流量本身的安全群組成員。這是 EFA 正常運作的必要項目。

在受限環境中使用 VPC 端點

在高度受限的環境中，您可能想要在沒有網際網路存取的情況下部署在子網路 AWS ParallelCluster 中。在這種情況下，您需要設定 VPC 端點，以允許叢集與 AWS 服務通訊。如需詳細說明，請參閱 [AWS ParallelCluster 在沒有網際網路存取的單一子網路中](#)。

使用 VPC 端點時，請確定您的安全群組允許往返 VPC 端點的流量。您可以將與 VPC 端點相關聯的安全群組新增至頭部節點和運算節點的 `AdditionalSecurityGroups` 組態，以執行此操作。

```
HeadNode:
...
Networking:
  SubnetId: subnet-1234567890abcdef0
  AdditionalSecurityGroups:
    - sg-abcdef01234567890 # Security group that enables communication with VPC
    endpoints
...

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
        - subnet-1234567890abcdef0
```

```
AdditionalSecurityGroups:
  - sg-1abcdef01234567890 # Security group that enables communication with VPC
    endpoints
```

安全群組組態的最佳實務

在受限環境中設定 的安全群組 AWS ParallelCluster 時，請考慮下列最佳實務：

- 最低權限原則：僅開啟叢集操作所需的連接埠。
- 使用安全群組參考：盡可能使用安全群組參考（允許來自另一個安全群組的流量），而不是 CIDR 區塊來限制叢集元件之間的流量。
- 限制 SSH 存取：將前端節點的 SSH 存取限制為僅使用 [HeadNode// 組態所需的 IP SshAllowedIps](#) 範圍。
- 限制 DCV 存取：如果使用 NICE DCV，請使用 [HeadNode// 組態僅限制存取需要的 IP DcvAllowedIps](#) 範圍。
- 徹底測試：設定自訂安全群組之後，請徹底測試所有叢集功能，以確保所有必要的通訊路徑都能正常運作。
- 記錄您的組態：維護安全群組組態的文件，包括開啟哪些連接埠，以及為什麼需要這些連接埠。

對安全群組問題進行故障診斷

如果您在設定自訂安全群組之後遇到問題，請考慮下列疑難排解步驟：

- 檢查叢集日誌：檢閱 CloudWatch Logs 中的叢集日誌是否有任何連線錯誤。
- 驗證安全群組規則：確保在適當的節點之間開啟所有必要的連接埠。
- 測試連線能力：使用 telnet 或 等工具 nc 來測試特定連接埠上節點之間的連線能力。
- 暫時展開規則：如果您無法識別需要哪些連接埠，請暫時允許叢集節點之間的所有流量，然後在識別所需的連接埠時逐漸限制。
- 檢查 VPC 端點組態：如果您使用的是 VPC 端點，請確保它們已正確設定，且安全群組允許進出流量。

如果持續遇到問題，您可以從叢集組態檔案移除 SecurityGroups 組態 AWS ParallelCluster，以還原為使用 建立的預設安全群組。

支援 AWS 區域 AWS ParallelCluster

AWS ParallelCluster 第 3 版可在下列內容中使用 AWS 區域：

區域名稱	區域	支援區域的第一個 AWS ParallelCluster 版本
美國東部 (俄亥俄)	us-east-2	3.0.0
美國東部 (維吉尼亞北部)	us-east-1	3.0.0
美國西部 (加利佛尼亞北部)	us-west-1	3.0.0
美國西部 (奧勒岡)	us-west-2	3.0.0
Africa (Cape Town)	af-south-1	3.0.0
亞太區域 (香港)	ap-east-1	3.0.0
亞太區域 (孟買)	ap-south-1	3.0.0
亞太區域 (首爾)	ap-northeast-2	3.0.0
亞太區域 (新加坡)	ap-southeast-1	3.0.0
亞太區域 (雪梨)	ap-southeast-2	3.0.0
亞太區域 (雅加達)	ap-southeast-3	3.10.0
亞太地區 (馬來西亞)	ap-southeast-5	3.13.0
亞太區域 (泰國)	ap-southeast-7	3.13.0
亞太區域 (東京)	ap-northeast-1	3.0.0
加拿大 (中部)	ca-central-1	3.0.0
中國 (北京)	cn-north-1	3.0.0
中國 (寧夏)	cn-northwest-1	3.0.0

區域名稱	區域	支援區域的第一個 AWS ParallelCluster 版本
歐洲 (法蘭克福)	eu-central-1	3.0.0
歐洲 (愛爾蘭)	eu-west-1	3.0.0
歐洲 (倫敦)	eu-west-2	3.0.0
歐洲 (米蘭)	eu-south-1	3.0.0
Europe (Paris)	eu-west-3	3.0.0
Europe (Stockholm)	eu-north-1	3.0.0
Middle East (Bahrain)	me-south-1	3.0.0
南美洲 (聖保羅)	sa-east-1	3.0.0
AWS GovCloud (美國東部)	us-gov-east-1	3.0.0
AWS GovCloud (美國西部)	us-gov-west-1	3.0.0
以色列 (特拉維夫)	il-central-1	3.8.0

版本備註和文件歷史記錄

下表說明 AWS ParallelCluster 使用者指南的主要更新和新功能。我們也會經常更新文件，以處理您傳送給我們的意見回饋。

AWS ParallelCluster

變更	描述	Date
AWS ParallelCluster 3.15.0 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>增強功能：</p> <ul style="list-style-type: none"> • 新增對 AL2 以外所有 OSs 的 p6-b300 執行個體支援。 • 以系統化計時器取代運算節點中的 <code>cfn-hup</code>，以支援就地更新，以改善大規模緊密耦合工作負載的效能。 • 停用 <code>dnf-makecache.timer</code> 以大規模改善 RHEL/Rocky 上緊密耦合工作負載的效能。 • 在叢集更新 Tags 期間支援的更新。 • 將 <code>LaunchTemplateOverrides</code> 新增至叢集組態，允許透過覆寫運算資源的啟動範本來自訂網路介面。 • 在缺少 <code>clustermgtd</code> 活動訊號時新增警示。 <p>變更：</p>	2026 年 3 月 23 日

變更	描述	Date
	<ul style="list-style-type: none">• 啟用 EFA 時，AWS ParallelCluster 現在會將網路介面設定為 interface 和 ，efa-only 而不是組合 efa 類型。這將 IP 地址使用量從每個網路卡一個減少到每個執行個體一個。• 升級 Slurm 至 25.11.4 版 (從 24.11.7)。• 將 Pmix 升級到 5.0.10 (從 5.0.6)。• 將 EFA 安裝程式升級至 1.47.0 (從 1.44.0)。• 將所有 OSs 的 NVIDIA 驅動程式升級至版本 580.105.08 (從 570.172.08) ， Amazon Linux 2 除外。• 將 GDRCopy 升級至 2.5.2 版 (從 2.4.4 版)。• 將 DCV 升級至 2025.0-20103 版 (從 2024.0-19030)。• 將 CUDA Toolkit 升級至 Amazon Linux 2 以外的所有 OSs 的 13.0.2 版 (自 12.8.1 起)。• 將 Python 升級至 3.14.2 (從 3.12.11 開始) ，適用於 Amazon Linux 2 以外的所有 OSs。• 將 Cinc 用戶端升級至 18.8.54 版 (從 18.7.10)。	

變更	描述	Date
	<ul style="list-style-type: none">• 將 Intel MPI Library 升級至 2021.17.2 (自 2021.16.0 起)。• 將 DCGM 升級至 Amazon Linux 2 以外的所有 OSs 4.5.1 版 (從 4.4.1 版)。• 將 Munge 升級至 Amazon Linux 2 以外的所有 OSs 0.5.17 版 (從 0.5.16 版)。• 透過在重試時重設中繼資料，減少 RHEL 和 Rocky out-of-sync 儲存庫鏡射所造成的暫時性建置映像失敗。• 無論失敗條件為何，叢集更新和運算機群狀態更新失敗時一律啟動 clustermgtd。• 改善叢集更新復原工作流程的彈性。• 將 Amazon Linux 2 以外的所有 OSs 的 NVIDIA Fabric Manager 升級至 580.105.08。• 將 aws-cfn-bootstrap 升級至 2.0-38 版 (從 2.0-33 版)。• 將所有 OSs 的 mysql-community-client 升級至 8.4.8 版 (從 8.0.39 版)，Amazon Linux 2 除外。• 將 Amazon Linux AMI 的 amazon-efs-utils 升級至 2.4.0 版 (自 v2.1.0 起)。	

變更	描述	Date
	<ul style="list-style-type: none">將 jmespath 升級至 ~=1.0 (從 ~=0.10)。升級表格至 <=0.9.0 (從 <=0.8.10)。新增驗證器以警告，以防在運算和登入節點上停用就地更新 (透過 DevSettings)。 <p>錯誤修正：</p> <ul style="list-style-type: none">使用具有隱含主路由表關聯的公有子網路時，無法公開存取修復 LoginNodes NLB。修正在建立具有 GPU 執行個體且已啟用 DCV 但無法存取網際網路的叢集時失敗的問題。修正當 head/compute/login 節點共用相同的安全群組時，叢集建立會因最終一致性而間歇性失敗的問題。在具有過時作業系統套件的 Ubuntu 父系映像上安裝 ubuntu-desktop 期間修正建置映像失敗。修正的驗證HeadNode/LocalStorage 。此組態參數不支援更新。修正驗證器PlacementGroupCapacityReservationValidator ，以	

變更	描述	Date
	<p>接受跨帳戶置放群組的容量保留。</p> <ul style="list-style-type: none">• 修正 CloudWatch 代理程式組態，以確保正確剖析所有日誌檔案的時間戳記。• 修正記錄組態以擷取所有 Slurm 運作狀態檢查事件 (將日誌層級從 WARNING 更新為 INFO，以防止遺失日誌項目)。• 確保更新不會在更新期間完成引導的節點上失敗，藉此改善叢集更新彈性。• 防止叢集更新失敗復原程序在 AWS 批次叢集上執行。此復原機制應該只在 Slurm 叢集上執行。 <p>棄用：</p> <ul style="list-style-type: none">• 不再支援自 3.14.0 起棄用的 LoginNodes/Pool/s/ssh/KeyName 組態參數。• 這是支援 Amazon Linux 2 的最後一個 AWS ParallelCluster 版本，因為 Amazon Linux 2 將於 2026 年 6 月 30 日終止支援。• 這是支援 AWS Batch CLI 的最後一個 AWS ParallelCluster 版本。從 v3.16.0 開始，AWS ParallelCluster 將	

變更	描述	Date
	不再支援 AWS Batch 做為排程器。	
AWS ParallelCluster 3.14.2 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>變更：</p> <ul style="list-style-type: none">• 升級 munge 至 0.5.18 版（從 0.5.16 版）以解決 CVE-2026-25506 的問題。• 將安裝程式中的 NodeJS 版本升級至 22.22.0 版（自 20.18.3 起）。	2026 年 2 月 16 日

變更	描述	Date
AWS ParallelCluster 3.14.1 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>變更：</p> <ul style="list-style-type: none">• 透過確保 <code>clustermgtd</code> 在更新成功完成後或在佇列重新設定成功失敗的更新後啟動，改善叢集更新彈性。• 新增 Chef 屬性 <code>cluster/in_place_update_on_fleet_enabled</code> 以停用運算和登入節點上的就地更新，並大規模降低效能影響。• 升級 Slurm 至 24.11.7 版 (從 24.11.6)。• 升級 Werkzeug 到 <code>~=3.1</code> (從 <code>~=2.0</code>) 以解決 CVE-2024-34069。• 將 Connexion 升級到 <code>~=2.15.1</code> (從 <code>~=2.13.0</code>)。• 升級 Flask 到 <code>~=3.1.0</code> (從 <code>>=2.2.5, <2.3</code>)。• 如果核心上有提供，請在安裝 NVIDIA 驅動程式 <code>drm_client_lib</code> 之前載入核心模組。• 安裝套件 <code>sssd-common</code>，而非 <code>sssd</code>，以減少相依性使用量 <code>sssd</code>。	2025 年 12 月 22 日

變更	描述	Date
	<ul style="list-style-type: none"> • 將 libjwt 升級至 Amazon Linux 2 以外所有OSes的 1.18.4 版 (從 1.17.0 版)。 • 將 amazon-efs-utils 升級至 2.4.0 版 (自 v2.3.1 起)。 • 將 EFA 安裝程式升級至 1.44.0 (從 1.43.2)。 • Efa-driver : efa-2.17.3-1 • Efa-config : efa-config-1.18-1 • Efa-profile : efa-profile-1.7-1 • Libfabric-aws : libfabric-aws-2.3.1-1 • Rdma-core : rdma-core-59.0-1 • 開啟 MPI : openmpi40-aws-4.1.7-2 和 openmpi50-aws-5.0.8-11 <p>錯誤修正：</p> <ul style="list-style-type: none"> • 修正 cfn-hup 在復原至超過 24 小時的叢集狀態後，因 cfn-signal 無法發出過期等待條件控制代碼的訊號，而在前端節點上進入無限迴圈的問題。 • 修正運算節點在更新失敗後可能部署錯誤叢集組態版本的競爭條件。 	

變更	描述	Date
	<ul style="list-style-type: none">防止在檢查進行時啟動執行個體導致叢集整備性檢查失敗。修正 CloudWatch Agent 組態中 chef-client.log 的不正確時間戳記剖析。在建置映像期間停用 Ubuntu 上的快照自動重新整理，以防止間歇性重新啟動失敗。僅保留已安裝核心的核心套件，將 Ubuntu 的 EFA 安裝時間縮短約 20 分鐘。將 GetFunction 和 GetPolicy 許可新增至 PClusterBuildImage CleanupRole，以防止建置映像堆疊刪除期間 AccessDenied 發生錯誤。當 DevSettings 為 null 或 DevSettings/InstanceTypesData 缺少必要欄位時，修正驗證錯誤訊息。	

變更	描述	Date
AWS ParallelCluster 3.14.0 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 包含 P6e-GB200 和 P6-B200 執行個體的驅動程式。ParallelCluster 會設定 Slurm 拓撲外掛程式來處理 P6e-GB200 UltraServers。如需重要的其他設定需求，請參閱限制一節。• 支援 <code>prioritized</code> 和 <code>capacity-optimized-prioritized</code> 配置策略。這可讓使用者排定放置執行個體的子網路優先順序，以最佳化成本和效能。• 根據核心 6.12 新增對 Amazon Linux 2023 AMIs <code>build-image</code> 支援（除了 6.1 之外）。• 在 Amazon Linux 2023 上支援 DCV。• 當節點無法引導時，執行個體主控台會顯示 Echo Chef-client 日誌。這有助於在 CloudWatch 日誌無法使用時調查引導失敗。 <p>限制：</p>	2025 年 9 月 30 日

變更	描述	Date
	<ul style="list-style-type: none"> • P6e-GB200 執行個體僅在 Amazon Linux 2023、Ubuntu 22.04 和 Ubuntu 24.04 上進行測試。 • 在 P6e-GB200 上使用 IMEX 需要額外設定。請參閱我們公有文件中的專用教學課程。 • P6-B200 執行個體僅在 Amazon Linux 2023、RHEL 8 和 9、Rocky 8 和 9、Ubuntu 22.04 和 Ubuntu 24.04 上進行測試。 • GPU HealthChecks 不建議用於 GPU 記憶體超過 320GB 的執行個體（例如 p6-b200.48xlarge）。運作狀態檢查持續時間可能超過 10 分鐘，可能導致任務失敗並大幅降低任務輸送量。 <p>變更：</p> <ul style="list-style-type: none"> • 為 Amazon Linux 2 以外的所有 OSs 安裝 nvidia-imex。 • UnkillableStepTimeout 從 slurm.conf 移除，並讓 slurm 設定此值。 • Lambda 函數使用的 Python 執行期升級至 Python 3.12（從 3.9 版）。如需 Python 3.9 EOL 的重要資訊，請參閱 Lambda 文件。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 支援透過新的組態參數 加密用於前端節點內部共用儲存體的 EFS 檔案系統HeadNode/SharedStorageEfsSettings/Encrypted 。 • 新增可警告 搭配 DCV 使用非 GPU 執行個體的驗證器。 • 升級 Slurm 至 24.11.6 版 (從 24.05.8)。 • 將 EFA 安裝程式升級至 1.43.2 (從 1.41.0 開始)。 <ul style="list-style-type: none"> • Efa-driver : efa-2.17.2-1 • Efa-config : efa-config-1.18-1 • Efa-profile : efa-profile-1.7-1 • Libfabric-aws : libfabric-aws-2.1.0-5 • Rdma-core : rdma-core-58.0-1 • 開啟 MPI : openmpi40-aws-4.1.7-2 和 openmpi50-aws-5.0.6-11 • 將 Cinc 用戶端升級至 18.4.12 版 (從 18.2.7)。 • 將所有OSs的 NVIDIA 驅動程式升級至版本 570.172.08 (從 570.86.15) , Amazon Linux 2 除外。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 將所有OSs的 CUDA Toolkit 升級到 12.8.1 版 (從 12.8.0 版) , Amazon Linux 2 除外。 • 將 DCGM 升級至 Amazon Linux 2 以外的所有OSs 4.4.1 版 (從 3.3.6 版) 。 • 將 Python 升級至 3.12.11 (從 3.12.8 版) , 適用於 Amazon Linux 2 以外的所有 OSs。 • 將 Python 升級至 Amazon Linux 2 的 3.9.23 (從 3.9.20)。 • 將 Intel MPI Library 升級至 2021.16.0 (自 2021.13.1 起) 。 • 將 DCV 升級至 2024.0-19 030 版。 • 將官方 ParallelCluster Amazon Linux 2023 AMIs 升級到核心 6.12 (從 6.1)。 <p>錯誤修正 :</p> <ul style="list-style-type: none"> • 透過部署全域角色 , 在映像成功或建置失敗後自動刪除build-image 堆疊 , 以防止堆疊build-image 刪除失敗。即使堆疊已刪除 , 該角色仍應存在。請參閱 https://github.com/aws/aws-parallelcluster/issues/5914。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 修正當規則同時包含 IPv4 範圍 (IpRanges) 和安全群組參考 (UserIdGroupPairs) 時，安全群組驗證失敗的問題。 • 修正 Rocky 9 上的 build-image 故障，當父映像未在最新的 Rocky 次要版本上運送最新的核心版本時發生。 • 修正使用 slurm 會計時導致叢集更新失敗的叢集 ID 不相符問題。 • 修正 CloudWatch Agent 啟動中可能導致節點引導失敗的競爭條件。 <p>棄用：</p> <ul style="list-style-type: none"> • 組態參數 LoginNodes/ Pools/Ssh/KeyName 已棄用，並在未來版本中移除。CLI 現在會在叢集組態中使用時傳回警告訊息。請參閱 https://github.com/aws/aws-parallelcluster/issues/6811。 • 不再支援 Ubuntu 20.04。 	

變更	描述	Date
AWS ParallelCluster 3.13.2 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>錯誤修正：</p> <ul style="list-style-type: none">修正當運算資源參考不再透過 EC2 API 存取的過期容量保留時，可能導致 <code>update-cluster</code> 和 <code>update-compute-fleet</code> 失敗的錯誤。APIs修正當父映像未運送最新的核心版本時，Rocky 9 上發生的 <code>build-image</code> 故障。請參閱 https://github.com/aws/aws-parallelcluster/issues/6874。	2025 年 6 月 24 日

變更	描述	Date
AWS ParallelCluster 3.13.1 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>變更：</p> <ul style="list-style-type: none">• 升級 Slurm 至 24.05.8 版。• 將 EFA 安裝程式升級至 1.41.0 (從 1.38.1 版)。<ul style="list-style-type: none">• Efa-driver : efa-2.15.0-1• Efa-config : efa-config-1.18-1• Efa-profile : efa-profile-1.7-1• Libfabric-aws : libfabric-aws-2.1.0-1• Rdma-core : rdma-core-57.0-1• 開啟 MPI : openmpi40-aws-4.1.7-2 和 openmpi50-aws-5.0.6• 將非 Amazon Linux AMI 的 amazon-efs-utils 升級至 2.3.1 版 (從 v2.1.0)。• 在 us-isob-east-1 和 us-iso-east-1 中支援 DCV。• 在 us-isob-east-1 和 us-iso-east-1 中支援 FSX for Lustre 和 Ontap。• 確保 ParallelCluster 映像建置的核心一致性，方法是在	2025 年 6 月 4 日

變更	描述	Date
	<p>開始時鎖定，並在完成時取消鎖定。</p> <p>錯誤修正：</p> <ul style="list-style-type: none">• 修正 ARM Performance Library 安裝中的錯誤，該錯誤導致在隔離環境中建置映像失敗。• 修正阻止指令碼 'update_directory_service_password.sh' 更新 AD 密碼的錯誤。	

變更	描述	Date
AWS ParallelCluster 3.13.0 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>棄用：</p> <ul style="list-style-type: none">這是支援 Ubuntu 20.04 的 ParallelCluster 版本，因為 Ubuntu 20.04 將於 2025 年 5 月進入 End-Of-Standard-Support。 <p>增強功能：</p> <ul style="list-style-type: none">新增對 Ubuntu 24.04 的支援。新增對 ap-southeast-7 區域的支援。從官方 ParallelCluster AMIs 停用未使用的服務杯和 <code>wpa_supplicant</code>，以提高安全性。 <p>變更：</p> <ul style="list-style-type: none">升級 Slurm 至 24.05.7 版。將所有 OSs 的 NVIDIA 驅動程式升級至版本 570.86.15 (從 550.127.08)，AL2 除外。將所有 OSs 的 CUDA Toolkit 升級至 12.8.0 版 (自 12.4.1 起)，AL2 除外。	2025 年 4 月 1 日

變更	描述	Date
	<ul style="list-style-type: none"> • 將 Python 升級至 3.12.8 , 適用於所有OSs但 AL2 除外 (自 3.9.20 起) 。 • 在 Ubuntu 22.04 上 , 使用用於編譯核心的相同編譯器版本安裝 Nvidia 驅動程式。 • aws-cfn-bootstrap 升級至 2.0-33 版。 • 將 EFA 安裝程式升級至 1.38.0 (從 1.36.0)。 • Efa-driver : efa-2.13.0-1 • Efa-config : efa-config-1.17-1 • Efa-profile : efa-profile-1.7-1 • Libfabric-aws : libfabric-aws-1.22.0-1 • Rdma-core : rdma-core-54.0-1 • 開啟 MPI : openmpi40-aws-4.1.7-1 和 openmpi50-aws-5.0.5 • 將 amazon-efs-utils 升級至 2.1.0 版。 • 移除第三方技術指南 : apt-7.5.22 和 pyenv-4.2.3。 • 升級第三方技術指南相依性 : <ul style="list-style-type: none"> • line-4.5.21 (從 line-4.5.13 起) 	

變更	描述	Date
	<ul style="list-style-type: none">• nfs-5.1.5 (來自 nfs-5.1.2)• openssh-2.11.14 (從 openssh-2.11.12 開始)• yum-7.4.20 (從 yum-7.4.13 起)• yum-epel-5.0.8 (來自 yum-epel-5.0.2)• 將 Pmix 升級到 5.0.6 (從 5.0.3)。• 將 ARM PL 升級到 24.10 版 (從 23.10 版)。• 在 Lambda layer 和安裝程式中將 Python 升級至 3.12.8 版 (自 3.9.17 起)。• 升級 NodeJS 至 Lambda layer 和安裝程式中的 20.18.3 版 (自 18.20.3 起)。• 將登入節點的 DSA 金鑰產生移除為 DSA，這會在 OpenSSH 9.7+ 中變成不支援。• 在運算節點啟動時，在 Slurm 中設定執行個體 ID 和執行個體類型資訊。• 在沒有選項 'no-cc-version-check' 的情況下安裝 NVIDIA 驅動程式，現已在 NVIDIA 安裝程式中棄用。• 新增驗證器以強制執行最多 10 個登入節點集區。	

變更	描述	Date
	<ul style="list-style-type: none">• 將預設根磁碟區大小更新為 45 GB。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 移除用於運算節點引導的 cfn-init 用量，以減少節點擴展時間。• 修正使用代理時導致運算節點引導失敗的問題。• 在 Ubuntu 22.04 上，使用與編譯核心相同的編譯器版本來安裝 Nvidia 驅動程式，以防止安裝失敗。- 修正在更新期間僅在主節點上覆寫 aws-parallelcluster-node 套件的執行。• 修正多使用者環境（與 Active Directory 整合）中透過 Pyxis/Enroot 執行的容器化任務會失敗的問題。• 修正使用目錄服務時，在 Rocky 9.5+ 上造成節點引導失敗的 authselect 用量。	

變更	描述	Date
AWS ParallelCluster 3.12.0 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code>。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 新增建置映像組態區段 <code>Build/Installation</code> 以開啟/關閉 NVIDIA 軟體和 Lustre 用戶端安裝。根據預設，雖然包含在官方 ParallelCluster AMIs 中，但不會安裝 NVIDIA 軟體 <code>build-image</code>。預設會安裝 Lustre 用戶端。• CLI 命令 <code>export-cluster-logs</code> 和 <code>export-image-logs</code> 現在可以根據預設將日誌匯出到預設的 ParallelCluster 儲存貯體，或如果在組態中指定，則匯出到 <code>CustomS3Bucket</code>。• 在 ARM 執行個體上將 Amazon DCV 支援擴展至 Ubuntu2204。 <p>變更：</p> <ul style="list-style-type: none">• 將 NVIDIA 驅動程式升級至 550.127.08 版（從 550.90.07）。這可解決 NVIDIA 的已知問題。如需詳細資訊，請參閱 NVIDIA 資料中心文件中的 已知問題。	2024 年 12 月 19 日

變更	描述	Date
	<ul style="list-style-type: none">• 將 Amazon DCV 升級到版本 2024.0-18131 。• 伺服器：2024.0-18131-1• xdcv：2024.0.631-1• gl：2024.0.1078-1• web_viewer： 2024.0-18131-1• 將 EFA 安裝程式升級至 1.36.0。• Efa-driver：efa-2.13.0-1• Efa-config：efa-config-1.17-1• Efa-profile：efa-profile-1.7-1• Libfabric-aws： libfabric-aws-1.22.0-1• Rdma 核心：rdma-core-54.0-1• 開啟 MPI：openmpi40-aws-4.1.7-1 和 openmpi50-aws-5.0.5• 失敗時自動重新啟動 slurmctld。• 將 mysql-community-client 升級到 8.0.39 版。• 移除對 Python 3.7 和 3.8 的支援，這是生命週期結束。	

變更	描述	Date
	<p>錯誤修正：</p> <ul style="list-style-type: none">• 修正叢集更新期間未偵測到自訂動作指令碼序列變更的問題。• 新增 AWS ParallelCluster API 缺少的許可，為部署登入節點所需的 Elastic Load Balancing 和 Auto Scaling 建立服務連結角色。• 修正在管理磁碟區時，我們取得區域的方式問題，讓它可以正確處理本機區域。• 修正在更新AccessPointIds 期間使用 新增 EFS 檔案系統失敗的問題。• 修正使用 PCAPI 時，叢集更新可能會在更新非 類型的參數時失敗的問題 String (例如，MaxCount)。• 掛載外部 OpenZFS 時，不再需要設定連接埠 111、2049、20001、20002、20003 的傳出規則。	

變更	描述	Date
AWS ParallelCluster 3.11.1 版已發行	<p>功能：</p> <ul style="list-style-type: none">• 依預設，Pyxis 現在已停用，因此必須如產品文件中所述手動啟用。• 在 ParallelCluster Lambda Layer 中將 Python 執行時間升級至 3.12 版。• 將 setuptools 的版本鎖定移除至 70.0.0 之前的版本。• 將 libjwt 升級至 1.17.0 版。• 完整變更日誌 <p>錯誤修正</p> <ul style="list-style-type: none">• 修正在 ParallelCluster 中設定 Pyxis Slurm 外掛程式時可能導致任務提交失敗的問題。• 透過在政策的公有範本中新增登入節點所需的缺少許可，修正導致在具有登入節點的組態中部署失敗的問題。https://github.com/aws/aws-parallelcluster/issues/6483	2024 年 10 月 21 日

變更	描述	Date
AWS ParallelCluster 3.11.0 版已發行	<p>增強功能</p> <ul style="list-style-type: none"> • 新增對登入節點上自訂動作的支援。 • 允許 DCV 連線登入節點。 • 新增對 ap-southeast-3 區域的支援。 • 將安全群組新增至登入節點網路負載平衡器。 • 新增登入節點的AllowedIps 組態。 • 新增組態SharedStorage/EfsSettings/AccessPointId 以指定掛載的選用 EFS 存取點 • 允許最多 10 個登入節點集區。 • 在官方 pcluster AMIs <p>變更</p> <ul style="list-style-type: none"> • 【中斷】 API DescribeCluster 和 CLI 命令傳回loginNodes 的欄位describe-cluster 已從字典變更為陣列，以支援多個登入節點集區。此變更會破壞回溯相容性，使這些操作與使用舊版部署的叢集不相容。 • 升級 Slurm 至 23.11.10 (從 23.11.7)。 	2024 年 9 月 26 日

變更	描述	Date
	<ul style="list-style-type: none"> • 將 Pmix 升級到 5.0.3 (從 5.0.2)。 • 將 EFA 安裝程式升級至 1.34.0。 <ul style="list-style-type: none"> • Efa-driver : efa-2.10.0-1 • Efa-config : efa-config-1.17-1 • Efa-profile : efa-profile-1.7-1 • Libfabric-aws : libfabric-aws-1.22.0-1 • Rdma 核心 : rdma-core-52.0-1 • 開啟 MPI : openmpi40-aws-4.1.6-3 和 openmpi50-aws-5.0.3-11 • 將 NVIDIA 驅動程式升級至 550.90.07 版 (從 535.183.01)。 • 將 CUDA Toolkit 升級到 12.4.1 版 (從 12.2.2)。 • 將 Python 升級到 3.9.20 (從 3.9.19)。 • 將 Intel MPI Library 升級至 2021.13.1.769 (自 2021.12.1.8 起)。 <p>錯誤修正</p>	

變更	描述	Date
	<ul style="list-style-type: none"> • 修正驗證器，EfaPlacementGroupValidator 使其不建議在使用容量區塊時設定置放群組。 • 透過確保 FSx for Lustre 檔案系統是在安全群組規則之後建立，來修正偶爾的叢集建立失敗。 • 修正啟用置放群組時的叢集刪除失敗。 • 修正限制 SSH 存取時，登入節點標示為運作狀態不佳的問題。 • 修正 <code>, retrieve_supported_regions</code> 讓它可以取得正確的 S3 URL。 • 修正 <code>describe_images</code> 以使用分頁。 • 修正將預設 VPC 子網路指定至 <code>LoginNodes/Networking/SubnetIds</code> 時的 <code>No route tables found</code> 錯誤。 	
AWS ParallelCluster 3.10.1 版已發行	<p>錯誤修正</p> <ul style="list-style-type: none"> • 修正中國區域的映像建置失敗。 	2024 年 7 月 8 日

變更	描述	Date
AWS ParallelCluster 3.10.0 版已發行	<p>增強功能：</p> <ul style="list-style-type: none">• 新增組態區段 <code>Scheduling/SlurmSettings/ExternalSlurmdbd</code>，將叢集連線至外部 Slurmdbd。• 允許在隔離的網路中執行建置映像。• 新增對 Amazon Linux 2023 的支援。• 將的支援 <code>price-capacity-optimized</code> 新增為 <code>AllocationStrategy</code>。• 新增驗證器，以防止將置放群組與容量區塊搭配使用。 <p>變更：</p> <ul style="list-style-type: none">• 不再支援 CentOS 7。• 從 18.4.12 將 Cinc 用戶端升級至 18.2.7 版。• 升級 munge 至 0.5.16 版 (從 0.5.15 版)。• 將 Pmix 升級到 5.0.2 (從 4.2.9)。• 升級第三方技術指南相依性：<ul style="list-style-type: none">• apt-7.5.22 (從 apt-7.5.14)• openssh-2.11.12 (從 openssh-2.11.3 開始)	2024 年 6 月 27 日

變更	描述	Date
	<ul style="list-style-type: none"> • 移除第三方技術指南： selinux-6.1.12。 • 將 EFA 安裝程式升級至 1.32.0。 • Efa-driver：efa-2.8.0-1 • Efa-config：efa-config-1.16-1 • Efa-profile：efa-profile-1.7-1 • Libfabric-aws： libfabric-aws-1.21.0-1 • Rdma 核心：rdma-core-50.0-1 • 開啟 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 535.154.05)。 • 將 Python 升級到 3.9.19 (從 3.9.17)。 • 將 Intel MPI Library 升級到 2021.12.1.8 (從 2021.9.0.43482)。 <p>錯誤修正：</p> <ul style="list-style-type: none"> • 修正要設為 AutoExportPolicy 和AutoImportPolicy 	

變更	描述	Date
	<p>tPolicy 選用的資料儲存庫關聯組態。</p> <ul style="list-style-type: none"> 修正叢集刪除期間的問題，該問題現在會在執行個體處於關機或終止狀態時完成運算機群清除。這是為了避免具有較長終止週期之執行個體類型的叢集刪除失敗。 允許啟用 cloudwatch 儀表板，並在叢集組態的 Monitoring 區段中停用警示。 允許 ParallelCluster 自訂資源使用 抑制驗證程式 PclusterCluster/SuppressValidators 。 移除 <code>/etc/profile.d/pcluster.sh</code> 使其不會在每次使用者登入時執行 <code>cf_n_boots_trap_virtualenv</code>，也不會 <code>PATH</code> 環境變數中新增。 將回應 failures 中的欄位取代 <code>failureReason</code> 為 <code>FailureReason</code>，以修正 ParallelCluster API 規格 <code>DescribeCluster</code> 。 新增缺少的 CloudFormation 堆疊狀態來修正 ParallelCluster API 規格：<code>IMPORT_*REVIEW_IN_PROGRESS</code> 和 <code>UPDATE_FAILED</code> 。 	

變更	描述	Date
	<ul style="list-style-type: none"> 修正無法讓叢集更新包含傳輸中加密的 EFS 檔案系統的問題。 修正在 EFS 用於共用內部資料時，導致 slurmctld 和 slurmdbd 服務無法在前端節點重新啟動時重新啟動的問題。 在 Ubuntu 系統上，移除與來自 ParallelCluster 的組態衝突之 cloud-init 日誌檔案的預設 logrotate 組態。 使用 RHEL 8.10 或更新版本修正映像建置失敗。 	
<p>AWS ParallelCluster 3.9.3 版已發行</p>	<p>若要升級，請輸入 <code>sudo pip install --upgrade aws-parallelcluster</code></p> <p>功能：</p> <ul style="list-style-type: none"> 新增支援 FSx Lustre 做為中的共用儲存類型 <code>us-iso-east-1</code>。 <p>錯誤修正：</p> <ul style="list-style-type: none"> <code>cloud_dns</code> 從 Slurm 組態 <code>SlurmctldParameters</code> 中的 <code>cloud_dns</code> 中移除，以避免 Slurm 散發問題。 <p>這不是必要的，因為我們在執行個體啟動時設定 IP 地址。</p>	<p>2024 年 6 月 19 日</p>

變更	描述	Date
AWS ParallelCluster 3.9.2 版已發行	<p>功能：</p> <ul style="list-style-type: none">• 升級 Slurm 至 23.11.7 (從 23.11.4)。• 如需詳細資訊，請參閱 GitHub CHANGELOG 3.9.2上的。	2024 年 5 月 28 日
AWS ParallelCluster 3.9.1 版已發行	<p>若要升級，請輸入下列項目： <code>sudo pip install --upgrade aws-parallelcluster</code></p> <p>錯誤修正</p> <ul style="list-style-type: none">• 在 update-cluster 操作中卸載檔案系統時，移除共用儲存體掛載的遞迴刪除。	2024 年 4 月 11 日

變更	描述	Date
AWS ParallelCluster 3.9.0 版已發行	<p>若要升級，請輸入下列項目：<code>sudo pip install --upgrade aws-parallelcluster</code></p> <p>增強功能：</p> <ul style="list-style-type: none">• 新增組態參數 <code>DeploymentSettings/DefaultUserHome</code>，以允許使用者將預設使用者的主目錄移至 <code>/local/home</code> 而不是 <code>/home</code> (預設)。• 允許更新 <code>MinCount</code>、<code>MaxCountQueue</code> 和 <code>ComputeResource</code> 組態參數，而不需要停止運算機群。現在可以透過將 <code>Scheduling/SlurmSettings/QueueUpdateStrategy</code> 設定為 <code>TERMINATE</code> 來更新它們。只會 AWS ParallelCluster 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 <code>Efs</code>、<code>FsxLustre</code>、<code>FsxOntap</code>、<code>FsxOpenZfs</code> 和 <code>FileCache</code> 類型的外部共用儲存體，而無需取代運算和登入機群。• 新增對 RHEL9 的支援。• 新增透過 <code>build-image</code> 程序 <code>CustomAmi</code> 建立的	2024 年 3 月 5 日

變更	描述	Date
	<p>Rocky Linux 9 支援。目前不提供公有 official AWS ParallelCluster Rocky9 Linux AMI。</p> <ul style="list-style-type: none"> • CommunicationParameters 從自訂 Slurm Settings 拒絕清單中移除。 • 新增 DeploymentSettings/DisableSudoAccessForDefaultUser 參數，以在支援的 OSes 中停用預設使用者的 sudo 存取。 • ParallelCluster 建立的 FSx for Lustre 檔案系統變更：將 Lustre 伺服器版本變更為 2.15。 • 透過 ['cluster'] ['nvidia']['kernel_open'] 技術指南節點屬性，在建置 AMI 時，在開啟和關閉來源 Nvidia 驅動程式之間新增選擇的可能性。 • * 新增 clustermgtd 組態選項 ec2_instance_missing_max_count ，以允許最終 Amazon EC2 的可設定重試次數描述執行個體與執行執行個體的一致性。 <p>變更</p> <ul style="list-style-type: none"> • 升級 Slurm 至 23.11.4 (從 23.02.7)。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 將 NVIDIA 驅動程式升級至 535.154.05 版。 • 在 pcluster CLI 和 aws-parallelcluster-batch-cli 中新增對 Python 3.11、3.12 的支援。 • 使用 Amazon EC2 DescribeInstances 回應 NetworkCardIndex 清單中的網路卡索引建置網路介面，而不是循環超過 MaximumNetworkCards 範圍。 • 使用執行個體類型 P3, G3, P2 和 G2 時，叢集建立失敗，因為其 GPU 架構與 3.8.0 版推出的開放原始碼 Nvidia 驅動程式 (OpenRM) 不相容。 • 升級第三方技術指南相依性：nfs-5.1.2 (從 nfs-5.0.0) • 升級 EFA 安裝程式至 1.30.0. <ul style="list-style-type: none"> • Efa-driver：efa-2.6.0-1 • Efa-config：efa-config-1.15-1 • Efa 設定檔：efa-profile-1.6-1 • Libfabric-aws：libfabric-aws-1.19.0 • Rdma 核心：rdma-core-46.0-1 	

變更	描述	Date
	<ul style="list-style-type: none"> • 開啟 MPI : openmpi40-aws-4.1.6-2 和 openmpi50-aws-5.0.0-11 • 將 NICE DCV 升級到 版本 2023.1-16388. • 伺服器 : 2023.1.16388-1 • xdcv : 2023.1.565-1 • gl : 2023.1.1047-1 • web_viewer : 2023.1.16388-1 <p>錯誤修正</p> <ul style="list-style-type: none"> • 修正從登入節點以 Active Directory 使用者身分提交時任務失敗的問題。此問題是由與前端節點上的外部 Active Directory 整合的不完整組態所造成。 • 重構 CloudFormation 範本 parallelclutser-policies.yaml 中定義的 IAM 政策，以防止超過 IAM 限制的政策導致 ParallelCluster API 部署失敗。 • 修正當前端節點寫入金鑰所需的時間超過預期時，導致登入節點無法引導的問題。 	

變更	描述	Date
	如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。	

變更	描述	Date
AWS ParallelCluster 3.8.0 版已發行	<p>AWS ParallelCluster 3.8.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 新增 ML 的 Amazon EC2 容量區塊支援。• 新增透過 build-image 程序 CustomAmi 建立的 Rocky Linux 8 支援。目前不提供公有 official AWS ParallelCluster Rocky8 Linux AMI。• 新增 Scheduling/ScalingStrategy 參數以控制啟動 Slurm 運算節點的 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，預設值 all-or-nothing 為。• 新增 HeadNode/SharedStorageType 參數以使用 EFS 儲存體，而不是從叢集內共用檔案系統資源的前端節點根磁碟區匯出 NFS：ParallelCluster、Intel、Slurm 和/home 資料。此增強功能可減少前端節點聯網的負載。• 允許透過 組態檔案的 SharedStorage 區段，將	2023 年 12 月 19 日

變更	描述	Date
	<p>掛/home載為 EFS 或 FSx 外部共用儲存體。</p> <ul style="list-style-type: none"> • 新增參數SlurmSettings/MungeKeySecretArn 以允許從 AWS Secrets Manager 使用外部使用者定義的 MUNGE 金鑰。 • 新增 Monitoring/Alarms/Enabled 參數以切換叢集的 Amazon CloudWatch 警示。 • 新增前端節點警示以監控 Amazon EC2 運作狀態檢查、CPU 使用率和前端節點的整體狀態，並將其新增至使用叢集建立的 CloudWatch Dashboard。 • 使用 DeploymentType PERSISTENT_2 做為受管 FSx for Lustre 時，新增對資料儲存庫關聯的支援。 • 新增 Scheduling/SlurmSettings/Database/DatabaseName 參數，以允許使用者為資料庫伺服器上要用於 Slurm 會計的資料庫指定自訂名稱。 • 在運算資源CapacityReservationTarget/CapacityReservationId 中設定時InstanceT 	

變更	描述	Date
	<p>type ，請建立選用的組態參數。</p> <ul style="list-style-type: none"> • 新增為 AWS ParallelCluster API 建立的 IAM 角色和政策指定字首的可能性。 • 新增可能以指定要套用至 AWS ParallelCluster API 建立之 IAM 角色和政策的許可界限。 <p>變更</p> <ul style="list-style-type: none"> • 升級 Slurm 至 23.02.7 (從 23.02.6)。 • 將 NVIDIA 驅動程式升級至 535.129.03 版。 • 將 CUDA Toolkit 升級到 12.2.2 版。 • 使用開放原始碼 NVIDIA GPU 驅動程式 (OpenRM) 做為 Linux 的 NVIDIA 核心模組，而非 NVIDIA 封閉式來源模組。 • 移除對 Slurm 恢復程式中 all_or_nothing_batch 組態參數的支援，以支持新的 Scheduling/ScalingStrategy 叢集組態。 • 將叢集警示命名慣例變更為 '【cluster-name】 - 【component-name】 - 【metric】'。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 針對根磁碟區和其他磁碟區，將 ADC 區域中的預設 EBS 磁碟區類型從 gp2 變更為 gp3。 • AWS ParallelCluster API 的選用許可界限現在會套用至 API 基礎設施建立的每個 IAM 角色。 <ul style="list-style-type: none"> • 將 EFA 安裝程式升級至 1.29.1。 • Efa-driver : efa-2.6.0-1 • Efa-config : efa-config-1.15-1 • Efa-profile : efa-profile-1.5-1 • Libfabric-aws : libfabric-aws-1.19.0-1 • Rdma 核心 : rdma-core-46.0-1 • 開啟 MPI : openmpi40-aws-4.1.6-1 • 在所有支援的OSes中將 GDRCopy 升級至 2.4 版，但使用 2.3.1 版的 Centos 7 除外。 • aws-cfn-bootstrap 升級至 2.0-28 版。 • 在 aws-parallelcluster-batchcli 中新增對 Python 3.10 的支援。 	

變更	描述	Date
	<p>錯誤修正</p> <ul style="list-style-type: none"> 修正在修改運算資源中宣告的執行個體類型清單時，叢集更新復原之後的不一致擴展組態。 修正在透過叢集組態檔案與外部 LDAP 伺服器整合的叢集中切換沒有根權限的使用者時產生 SSH 金鑰的問題。 修正在設定時停用 Slurm 省電模式的問題 Scaledown Idletime = -1 。 修正 Slurm 會計 update_slurm_database_password.sh 指令碼中 Slurm 安裝 dir 的硬式編碼路徑。 	
AWS ParallelCluster 3.7.2 版已發行	<p>AWS ParallelCluster 3.7.2 版已發行。</p> <p>變更：</p> <ul style="list-style-type: none"> 升級 Slurm 至 23.02.6。 	2023 年 10 月 25 日

變更	描述	Date
AWS ParallelCluster 3.7.1 版已發行	<p>AWS ParallelCluster 3.7.1 版已發行。</p> <p>變更：</p> <ul style="list-style-type: none">• 升級 Slurm 至 23.02.5 (從 23.02.4)。• 將 Pmix 升級到 4.2.6 (從 3.2.3)。• 將 libjwt 升級到 1.15.3 (從 1.12.0)。• 將 EFA 安裝程式升級至 1.26.1，修復 P5 中的 RDMA 寫入資料問題。<ul style="list-style-type: none">• Efa-driver : efa-2.5.0-1 。• Efa-config : efa-config-1.15-1 。• Efa-profile : efa-profile-1.5-1 。• Libfabric-aws : libfabric-aws-1.18.2-1 。• ERdma-core : rdma-core-46.0-1 。• 開啟 MPI : openmpi40-aws-4.1.5-4 。	2023 年 9 月 22 日

變更	描述	Date
AWS ParallelCluster 3.7.0 版已發行	<p>AWS ParallelCluster 3.7.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 使用組態 YAML 檔案支援運算資源中靜態和動態節點優先順序的 AWS ParallelCluster 組態。• 新增對 Ubuntu 22 的支援。預設不支援 RSA 金鑰。• 新增佇列組態設定 <code>JobExclusiveAllocation</code>，以在任何指定時間將分割區中的節點專門配置給單一任務。• 允許在叢集建立和叢集更新時間覆寫 <code>aws-parallelcluster-node</code> 套件。對於前端節點，這適用於叢集更新。僅適用於開發用途。• 避免 NFS 伺服器在運算節點上啟動。• 新增對登入節點的支援。• 為 Slurm Compute Resource 指定多個執行個體類型時，允許記憶體型排程。• 新增支援，將現有的 Amazon File Cache 掛載為共用儲存。	2023 年 8 月 30 日

變更	描述	Date
	<p>變更：</p> <ul style="list-style-type: none"> • 根據預設，指派 Slurm 動態節點的優先順序（權重）為 1000。透過這樣做，Slurm 可以優先考慮閒置靜態節點而非閒置動態節點。 • 讓aws-parallelcluster-node 協助程式只處理 AWS ParallelCluster 受管 Slurm 分割區。 • 將EFS-utils 監視程式輪詢間隔增加到 10 秒。此變更適用於 EncryptionInTransit 設定為 true，這是導致監視程式執行的唯一條件。 • 將 EFA 安裝程式升級至 1.25.1。 <ul style="list-style-type: none"> • Efa-driver：efa-2.5.0-1（來自 efa-2.1.1g） • Efa-config：efa-config-1.15-1（來自 efa-config-1.13-1） • Efa-profile：efa-profile-1.5-1（無變更） • Libfabric-aws：libfabric-aws-1.18.1-0（來自 libfabric-aws-1.17.1-1） 	

變更	描述	Date
	<ul style="list-style-type: none">• Rdma-core : rdma-core-46.0-1 (來自 rdma-core-43.0-1)• 開啟 MPI : openmpi40-aws-4.1.5-4 (來自 openmpi40-aws-4.1.5-1)• 升級 Slurm 至 23.02.4 版。• 將 lmds/lmdsSupport 的預設值從 v1.0 變更為 v2.0。• 棄用 Ubuntu 18。• 將預設根磁碟區大小更新為 40 GB，以考量 Centos 7 的限制。• 限制前端節點內檔案 /tmp/wait_condition_handle.txt 的許可，以便只有根才能讀取。• 建立節點套件常駐程式要使用的 Slurm 分割區節點清單映射 JSON 檔案，以識別 PC 管理的 Slurm 分割區和節點清單。• 將 NVIDIA 驅動程式升級至 535.54.03 版。• 將 CUDA 程式庫升級至 12.2.0 版。• 將 NVIDIA Fabric Manager 升級到 nvidia-fabricmanager-535。	

變更	描述	Date
	<ul style="list-style-type: none"> • 僅將 ARM PL 升級至 Ubuntu 22.04 的 23.04.1 版。 • 將 NICE DCV 升級至版本 2023.0-15487 。 • 伺服器：2023.0.15487-1 • xdcv：2023.0.551-1 • gl：2023.0.1039-1 • web_viewer：2023.0.15487-1 <p>錯誤修正：</p> <ul style="list-style-type: none"> • 將驗證新增至 Scaledown IdleTime 值，以防止將值設定為低於 -1。 • 在啟用 DCV 的 GPU 執行個體上，使用 Ubuntu Deep Learning AMI 修正叢集建立失敗。 • 修正使用 CustomLambdaRole 建立 ParallelCluster CloudFormation 自訂資源提供者時，導致建立懸置 IAM 政策的問題。 • 修正使用 SlurmSettings/Dns/UseEc2Hostnames 等於 True 時，在具有多個網路介面的執行個體上造成運算節點 DNS 名稱不一致的問題 True 	

變更	描述	Date
	如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster 、 aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。	
僅限文件版本	AWS ParallelCluster 已發佈第 3 版的特定使用者指南。 僅限文件版本： <ul style="list-style-type: none">• AWS ParallelCluster 第 3 版有自己的個別使用者指南。	2023 年 7 月 17 日

變更	描述	Date
AWS ParallelCluster 3.6.1 版已發行	<p>AWS ParallelCluster 3.6.1 版已發行。</p> <p>變更：</p> <ul style="list-style-type: none">• <code>clustermgtd</code> 如果運算節點新增至多個 Slurm 分割區，請避免看到的節點重複。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 移除根磁碟區裝置名稱 (<code>/dev/sda1</code> 和 <code>/dev/xvda</code>) 的硬式編碼，並從期間使用的 AMIs 中擷取 <code>create-cluster</code>。• 修正使用 CloudFormation 自訂資源並將 <code>ElasticIp</code> 設定為 <code>時的叢集建立失敗True</code>。• 修正叢集在搭配大型組態檔案使用 a AWS CloudFormation 自訂資源時的建立和更新失敗。• 修正無法在 Ubuntu 上停用 <code>ptrace</code> 保護，且不允許 <code>libfabric</code> 中跨記憶體連接 (CMA) 的問題。• 修正使用多種執行個體類型且沒有傳回執行個體時，容量快速不足的容錯移轉邏輯。	2023 年 7 月 5 日

變更	描述	Date
	如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster 、 aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。	

變更	描述	Date
AWS ParallelCluster 3.6.0 版已發行	<p>AWS ParallelCluster 3.6.0 版已發行。</p> <p>文件：</p> <ul style="list-style-type: none">• 新增 AWS ParallelCluster Python 程式庫 API 的文件。 <p>增強功能：</p> <ul style="list-style-type: none">• 新增對 RHEL8 的支援。• 新增 AWS CloudFormation 自訂資源，以使用 CloudFormation 建立和管理叢集。• 新增在 組態 YAML 檔案中自訂叢集 Slurm AWS ParallelCluster 組態的支援。• 建置支援 LUA 的 Slurm。• 將每個叢集的佇列數量上限從 10 個增加到 50 個。每個佇列最多可有 50 個運算資源。每個叢集最多可有 50 個運算資源。• 新增為 OnNodeStart、OnNodeConfigured 和 OnNodeUpdated 參數中設定的事件指定多個 自訂動作指令碼 序列的支援。• 新增組態區段 HealthChecks / Gpu，用於在執行任務之前在運算節點上套用 GPU 運作狀態檢查。	2023 年 5 月 22 日

變更	描述	Date
	<ul style="list-style-type: none"> • 在 SlurmQueues 和 / SlurmQueues ComputeResources 組態Tags中新增對 的支援。 • 在Monitoring 組態中新增對 DetailedMonitoring 的支援。 • 在 CloudWatch 儀表板中 AWS ParallelCluster 新增前端節點記憶體和根磁碟區磁碟使用率追蹤的 mem_used_percent 和 disk_used_percent 指標，並設定監控這些指標的警示。 • 新增 AWS ParallelCluster 受管日誌的日誌輪換支援。 • 在 CloudWatch Dashboard 中追蹤常見的運算節點錯誤和動態節點最長閒置時間。 • 建立 SSL Socket 時，強制 DCV Authenticator Server 至少使用TLS-1.2通訊協定。 • 在所有支援的作業系統上安裝 NVIDIA Data Center GPU Manager (DCGM) 套件，aarch64centos7和除外alinux2。 • 根據預設，載入核心模組 nvidia-vm，以提供統一虛擬記憶體 (UVM) 功能給 CUDA 驅動程式。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 安裝 NVIDIA 持續性協助程式 做為系統服務。 <p>變更：</p> <ul style="list-style-type: none"> • 升級 Slurm 至版本 23.02.2 (從版本 22.05.8)。 • 將 munge 升級到版本 0.5.15 (從版本 0.5.14)。 • 將 Slurm TreeWidth 設定為 30。 • 將 Slurm prolog和epilog組態/opt/slurm/etc/scripts/epilog.d/ 分別設定為目標目錄 /opt/slurm/etc/scripts/prolog.d/ 和 。 • 在運算節點註冊期間，將Prolog執行指令碼的 Slurm BatchStartTimeout 設定為 3 分鐘上限。 • 將 CloudWatch 日誌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。 • Dkms : 2.8.3-2 	

變更	描述	Date
	<ul style="list-style-type: none"> • Efa-driver : efa-2.1.1g (無變更) • Efa-config : efa-config-1.13-1 (無變更) • Efa-profile : efa-profile-1.5-1 (無變更) • Libfabric-aws : libfabric-aws-1.17.1-1 (來自 libfabric-aws-1.17.0-1) • Rdma-core : rdma-core-43.0-1 (無變更) • 開啟 MPI : openmpi40-aws-4.1.5-1 (無變更) • 在 Amazon Linux 2 2.12 上將 Lustre 用戶端版本升級至 。 Lustre 用戶端 2.12 已安裝在 Ubuntu 20.04、18.04 和 CentOS >= 7.7 上。 • 在 CentOS 7.6 2.10.8 上將 Lustre 用戶端版本升級至 。 • 將 NVIDIA 驅動程式升級至版本 470.182.03 (從版本)470.141.03 。 • 將 NVIDIA Fabric Manager 升級到版本 470.182.03 (從版本)470.141.03 。 • 將 NVIDIA CUDA Toolkit 升級到版本 11.8.0 (從版本)11.7.1。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 將 NVIDIA CUDA 範例升級至版本 11.8.0。 • 將 Intel MPI 程式庫升級至版本 2021 Update 9 (自版本 2021 Update 6 起)。如需詳細資訊，請參閱 Intel® MPI Library 2021 Update 9。 • 將 NICE DCV 升級到版本 2023.0-15022 (從版本 2022.2-14521)。 <ul style="list-style-type: none"> • 伺服器：2023.0.15022-1 (從版本)2022.2-14521-1 。 • xdcv：2023.0.547-1 (從版本)2022.2.519-1 。 • gl：2023.0.1027-1 (從版本)2022.2.1012-1 。 • web_viewer：2023.0.15022-1 (從版本開始2022.2.14521-1)。 • aws-cfn-bootstrap 升級至版本 2.0-24。 • 為 AWS 批次叢集建置容器映像時，CodeBuild 環境使用的升級映像： <ul style="list-style-type: none"> • aws/codebuild/amazonlinux2-x86_64-standard:4.0 (來自 aws/codebuild/amaz 	

變更	描述	Date
	<p>onlinux2-x86_64-standard:3.0)。</p> <ul style="list-style-type: none"> aws/codebuild/amazonlinux2-aarch64-standard:2.0 (來自aws/codebuild/amazonlinux2-aarch64-standard:1.0)。 <p>錯誤修正：</p> <ul style="list-style-type: none"> 修正 Amazon EFS 和 Amazon FSx 網路安全群組驗證器，以避免報告錯誤。 修正 Image Builder 在build-image 操作期間所建立資源的遺失標記。 修正的更新政策MaxCount，以一律在MaxCount 屬性上執行數值比較。 使用多個網路卡修正運算節點執行個體上的 IP 對齊。 在執行佇列參數更新且未更新 Slurm 會計組態slurm_parallelcluster_slurmdbd.conf 時，修正StoragePass 中的替換。 修正使用現有 EFS 檔案系統建立叢集時，導致建立懸置安全群組的問題。 	

變更	描述	Date
	<ul style="list-style-type: none">• 修正重新啟動時導致cfn-hup協助程式失敗的問題。• 將具有INVALID_REG 旗標的動態節點視為 Slurm 保護模式的引導失敗。未通過 Slurm 註冊的靜態節點已在之後視為引導失敗node_replacement_timeout 。 <p>如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 3.5.1 版已發行	<p>AWS ParallelCluster 3.5.1 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 新增獨立的 pcluster CLI 安裝程式可執行檔。 <p>變更：</p> <ul style="list-style-type: none">• 將 EFA 安裝程式升級至 1.22.0。<ul style="list-style-type: none">• Efa-driver : efa-2.1.1g (來自 efa-2.1.1-1)• Efa-config : efa-config-1.13-1 (來自 efa-config-1.12-1)• Efa-profile : efa-profile-1.5-1 (無變更)• Libfabric-aws : libfabric-aws-1.17.0-1 (來自 libfabric-aws-1.16.1amzn3.0-1)• Rdma-core : rdma-core-43.0-1 (無變更)• 開啟 MPI : openmpi40-aws-4.1.5-1 (來自 openmpi40-aws-4.1.4-3) <p>將 NICE DCV 升級至版本 2022.2-14521 。</p>	2023 年 3 月 29 日

變更	描述	Date
	<ul style="list-style-type: none">• 伺服器：2022.2.14521-1• xdcv：2022.2.519-1• gl：2022.2.1012-1• web_viewer：2022.2.14521-1 <p>錯誤修正：</p> <ul style="list-style-type: none">• 修正在叢集更新中移除共用 Amazon EBS 磁碟區/ etc/exports 時，因 MountDir和 之間的模式比對所導致的潛在節點啟動失敗。• 修正 以防止日誌compute_console_output 檔案在每次clustermgtd 反覆運算時截斷。 <p>如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 3.5.0 版已發行	<p>AWS ParallelCluster 3.5.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 使用 AWS ParallelCluster UI 存取和管理叢集。• 在 CloudFormation 範本中新增版本控制 AWS ParallelCluster 政策，供您在工作負載中參考。• 新增 Python AWS ParallelCluster 程式庫，以搭配您自己的程式碼使用。• 在運算節點引導失敗時，將運算節點主控台輸出的記錄新增至 Amazon CloudWatch。• 新增失敗欄位，其中包含當叢集建立失敗時要describe-cluster 輸出的失敗代碼和原因。• 新增驗證程式，以防止在呼叫子程序模組時插入惡意字串。• 佈建靜態節點PROTECTED時，如果叢集狀態變更為，則建立叢集失敗。 <p>變更：</p> <ul style="list-style-type: none">• 升級到 Slurm 版本 22.05.8 (從版本開始22.05.7)	2023 年 2 月 20 日

變更	描述	Date
	<ul style="list-style-type: none"> • 將 EFA 安裝程式升級至 1.21.0。 • Efa-driver : efa-2.1.1-1 (來自 efa-2.1) • Efa-config : efa-config-1.12-1 (來自 efa-config-1.11-1) • Efa-profile : efa-profile-1.5-1 (無變更) • Libfabric-aws : libfabric-aws-1.16.1amzn3.0-1 (來自 libfabric-aws-1.16.1) • Rdma-core : rdma-core-43.0-1 (來自 rdma-core-43.0-2) • 開啟 MPI : openmpi40-aws-4.1.4-3 (無變更) • 讓 Slurm 控制器更詳細地記錄，並啟用 Slurm 省電外掛程式的額外記錄。 <p>錯誤修正：</p> <ul style="list-style-type: none"> • 修正叢集資料庫建立，方法是在啟用 Slurm 會計時，驗證叢集名稱不超過 40 個字元。 • 修正中clustermgtd 導致運算節點在 Amazon EC2 執 	

變更	描述	Date
	<p>行個體狀態檢查失敗時透過 Slurm 重新啟動的問題。</p> <ul style="list-style-type: none"> 修正因為前端節點上的 IAM 政策不正確，導致其他帳戶共用容量保留無法啟動的運算節點問題。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook、aws-parallelcluster-r-node 和 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	
<p>AWS ParallelCluster 3.4.1 版已發行</p>	<p>AWS ParallelCluster 3.4.1 版已發行。</p> <p>錯誤修正：</p> <ul style="list-style-type: none"> 修正 Slurm 排程器問題，這些問題可能導致不正確地套用更新至其運算節點的內部登錄檔。因此，如果此問題，EC2 執行個體可能會變得無法使用，或可能受到不正確執行個體類型的支援。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	<p>2023 年 1 月 13 日</p>

變更	描述	Date
AWS ParallelCluster 3.4.0 版已發行	<p>AWS ParallelCluster 3.4.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 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 / ResourcePrefix 中新增組態參數，以指定由建立之 IAM 資源的路徑和名稱的字首 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。• 新增在叢集更新期間指定要在前端節點中執行的自訂指令碼的功能。使用 Slurm 做為排程器時，可以使用 HeadNode / CustomActions / OnNodeUpdated 指定指令碼。 <p>變更：</p> <ul style="list-style-type: none">• 移除為現有檔案系統建立 Amazon EFS 掛載目標。	2022 年 12 月 22 日

變更	描述	Date
	<ul style="list-style-type: none"> • 使用掛載 EFS 檔案系統amazon-efs-utils。EFS 檔案系統可以使用傳輸中加密和 IAM 授權使用者進行掛載。 • 在 CentOS7 和 Ubuntu 上安裝 stunnel 5.67，以支援 EFS 傳輸中加密。 • 將 EFA 安裝程式升級至 1.20.0 (從 1.18.0)。 <ul style="list-style-type: none"> • Efa-driver : efa-2.1 (來自 efa-1.16.0-1) • Efa-config : efa-config-1.11-1 (無變更) • Efa-profile : efa-profile-1.5-1 (無變更) • Libfabric-aws : libfabric-aws-1.16.1 (來自 libfabric-aws-1.16.0~amzn4.0-1) • Rdma-core : rdma-core-43.0-2 來自 (rdma-core-41.0-2) • 開啟 MPI : openmpi40-aws-4.1.4-3 來自 (openmpi40-aws-4.1.4-2) • 升級 Slurm 至版本 22.05.7 (從 22.05.5)。 	

變更	描述	Date
	<ul style="list-style-type: none">• 將 Python 升級到 3.9.16 和 3.7.16。 (從 3.9.15 和 3.7.13)。• 使用 Slurm 時 22.05.7，處於 IDLE+CLOUD+COMPLETING+POWER_DOWN+NOT_RESPONDING 狀態的動態節點不會被視為運作狀態不佳。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件 CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 3.3.1 版已發行	<p>AWS ParallelCluster 3.3.1 版已發行。</p> <p>變更：</p> <ul style="list-style-type: none"> 官方 AWS ParallelCluster 產品 AMIs 現已在兩年的 Amazon EC2 棄用後提供。 將 AWS ParallelCluster API Lambda 的記憶體大小增加至 2048，以減少冷啟動懲罰並避免逾時。 <p>錯誤修正：</p> <ul style="list-style-type: none"> 防止替換受管 FSx for Lustre 檔案系統，以及遺失包含運算機群子網路 ID 變更的叢集更新資料。 SharedStorage DeletionPolicy 適用於叢集更新動作。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster 套件CHANGELOG 的檔案。</p>	2022 年 12 月 2 日
AWS ParallelCluster 文件專用 hpc6id 備註	<p>AWS ParallelCluster 文件限定更新</p> <ul style="list-style-type: none"> AWS ParallelCluster 不支援 HeadNode / InstanceType 設定的 hpc6id 執行個體類型。 	2022 年 12 月 2 日

變更	描述	Date
AWS ParallelCluster 3.1.5 版已發行	<p>AWS ParallelCluster 3.1.5 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none"> • 修正防止閒置節點終止的 Slurm 問題。 • 將 EFA 安裝程式升級至 1.18.0 <ul style="list-style-type: none"> • Efa-driver : efa-1.16.0-1 • Efa-config : efa-config-1.11-1 (來自 efa-config-1.9-1) • Efa-profile : efa-profile-1.5-1 (無變更) • Libfabric-aws : libfabric-aws-1.16.0~amzn4.0-1 (來自 libfabric-1.13.2)。 • Rdma-core : rdma-core-41.0-2 (來自 rdma-core-37.0) • 開啟 MPI : openmpi40-aws-4.1.4-2 (來自 openmpi40-aws-4.1.1-2) <p>變更：</p> <ul style="list-style-type: none"> • ParallelClusterUserRole 將 lambda:ListTags 和 lambda:UntagResource 新增至 	2022 年 11 月 16 日

變更	描述	Date
	<p>AWS ParallelCluster API 堆疊用於叢集更新的。</p> <ul style="list-style-type: none">• 將 Intel MPI Library 升級至版本 2021 Update 6 (自版本 2021 Update 4 起)。如需詳細資訊，請參閱 Intel® MPI Library 2021 Update 6。• 將 NVIDIA 驅動程式升級至 470.141.03 版 (從 470.103.01)。• 將 NVIDIA Fabric Manager 升級到 470.141.03 版 (從 470.103.01)。 <p>如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 3.3.0 版已發行	<p>AWS ParallelCluster 3.3.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none"> • 使用 Slurm 做為排程器時，為運算資源新增對多個執行個體配置組態的支援。如需詳細資訊，請參閱使用 Slurm 進行多個執行個體類型配置。 • 使用更新的組態，新增對使用叢集更新新增和移除 SharedStorage 的支援。如需詳細資訊，請參閱共用儲存。 • 新增 Efs DeletionPolicy 和 FsxLustre 共用儲存設定的新組態參數，以支援儲存保留。 • 使用新的組態參數排程/SlurmSettings/資料庫新增對 Slurm 會計的支援。./Scheduling-v3.html#Scheduling-v3-SlurmSettings-Database如需詳細資訊，請參閱使用 Slurm 會計 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預留 (ODCR) 啟動執行個體。 	2022 年 11 月 2 日

變更	描述	Date
	<ul style="list-style-type: none"> • 新增組態參數以指定要在叢集中支援的 IMDS 版本，或在叢集中建置映像基礎設施、Imds / ImdsSupport 和建置、Imds / ImdsSupport、組態。 • 在 SlurmQueues / ComputeResources 區段中新增對網路 / Placement Group 的支援。 • 新增對具有多個網路介面的執行個體的支援，每個裝置僅限一個 ENI。 • 檢查已連接安全群組中的 CIDR 區塊，以改善外部 Amazon EFS 檔案系統的聯網驗證。 • 新增驗證器，以檢查設定的執行個體類型是否支援置放群組。 • 將 NFS 執行緒設定為 $\min(256, \max(8, \text{num_cores} * 4))$，以確保更好的穩定性和效能。 • 在建置時間移動 NFS 安裝，以減少組態時間。 • 針對部署 AWS ParallelCluster API 時建立的 EcrImageBuilder SNS 主題啟用伺服器端加密，並用於通知 docker 映像建置事件。 <p>變更：</p>	

變更	描述	Date
	<ul style="list-style-type: none">• 變更 SlurmQueues / Networking / Placement Group / 的行為Enabled。它現在會為每個運算資源建立唯一的受管置放群組，而不是為所有運算資源建立單一受管置放群組。• 新增對 SlurmQueues / Networking / Placement Group / Name 的支援作為偏好的命名方法。• 將前端節點標籤從啟動範本移至執行個體定義，以避免在標籤更新時更換前端節點。• 停用透過啟動範本中執行的指令碼執行的多執行緒cloud-init，而不是透過CpuOptions 設定。• 在 API 基礎設施、API Docker 容器和叢集 Lambda 資源中，將 Python 升級至 3.9 版和 NodeJS 升級至 16 版。• 在中移除對 Python 3aws-parallelcluster-batch-cli .6 的支援。• 升級 Slurm 至版本 22.05.5 (從 21.08.8-2)。• 將 NVIDIA 驅動程式升級至版本 470.141.03 (從 470.129.06)。	

變更	描述	Date
	<ul style="list-style-type: none"> • 將 NVIDIA Fabric Manager 升級到版本 470.141.03 (從 470.129.06)。 • 將 NVIDIA CUDA Toolkit 升級到 11.7.1 版 (from 11.4.4)。 • 將 AWS ParallelCluster Virtualenvs 中使用的 Python 從升級至 3.7.13 3.9.15。 • 將 EFA 安裝程式升級至 1.18.0 版。 <ul style="list-style-type: none"> • Efa-driver : efa-1.16.0-1 (無變更) • Efa-config : efa-config-1.11-1 (from efa-config-1.10-1) • Efa-profile : efa-profile-1.5-1 (無變更) • Libfabric-aws : libfabric-aws-1.16.0~amzn4.0-1 (來自 libfabric-aws-1.16.0~amzn2.0-1)。 • Rdma-core : rdma-core-41.0-2 (來自 rdma-core-37.0) • 開啟 MPI : openmpi40-aws-4.1.4-2 (來自 openmpi40-aws-4.1.1-2) 	

變更	描述	Date
	<ul style="list-style-type: none"> • 將 NICE DCV 升級至版本 2022.1-13300 (從 2022.0-12760)。 • 為 啟用抑制 SingleSubnetValidator Queues。 • 當 DRAIN 節點處於 COMPLETING 狀態時，請勿取代節點，因為 Epilog 可能仍在執行中。 <p>錯誤修正：</p> <ul style="list-style-type: none"> • 修正 AWS ParallelCluster ListClusterLogStreams 命令中篩選條件參數的驗證，以便在傳遞不正確的篩選條件時失敗。 • 修正當與其他 SharedStorage / EfsSettings 參數一起指定 FileSystemId 時，參數 SharedStorage / EfsSettings 的驗證失敗。先前 FileSystemId 並未包含。 • 修正變更 SharedStorage 順序與組態中其他變更時的叢集更新。 • 修正 AWS ParallelCluster API UpdateParallelClusterLambdaRole 中的，將日誌上傳至 CloudWatch。 	

變更	描述	Date
	<ul style="list-style-type: none"> 在執行任何技術指南之前安裝套件時，修正 Cinc 不使用本機 CA 憑證套件。 使用 <code>Build:UpdateOsPackages:Enabled:true</code> 設定 <code>pcluster build-image</code> 時，修正升級 ubuntu 時的懸置。 透過重複金鑰失敗來修正 YAML 叢集組態的剖析。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	
AWS ParallelCluster 僅新增文件 API 參考。	<p>AWS ParallelCluster 文件限定更新</p> <ul style="list-style-type: none"> 已將第 3 版 AWS ParallelCluster API 參考 新增至文件。 	2022 年 10 月 27 日

變更	描述	Date
AWS ParallelCluster 3.2.1 版已發行	<p>AWS ParallelCluster 3.2.1 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">改善邏輯，將主機路由表與不同的網路卡建立關聯，以更好地支援具有多個 NICs Amazon EC2 執行個體。 <p>變更：</p> <ul style="list-style-type: none">將 NVIDIA 驅動程式升級至 470.141.03 版。將 NVIDIA Fabric Manager 升級到 470.141.03 版。停用 cron 任務 man-db 和 mlocate，這可能會對節點效能產生負面影響。將 Intel MPI Library 升級至 2021.6.0.602。將 Python 從 3.7.10 升級到 3.7.13，以回應此安全風險。 <p>錯誤修正：</p> <ul style="list-style-type: none">避免在叢集組態無法使用 DescribeCluster 時失敗。 <p>如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster、aws-paral</p>	2022 年 10 月 3 日

變更	描述	Date
	lelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。	

變更	描述	Date
<p>AWS ParallelCluster 3.2.0 版已發行</p>	<p>AWS ParallelCluster 3.2.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none"> • 在 Slurm 中新增記憶體型排程的支援。 • 在 Slurm 叢集組態中設定運算節點實際記憶體。 • 新增組態參數排程 / SlurmSettings / EnableMemoryBasedScheduling，以在 Slurm 中啟用記憶體型排程。 • 新增組態參數 Scheduling / SlurmQueues / ComputeResources / SchedulableMemory，以覆寫排程器在運算節點上看到的記憶體預設值。 • 改善叢集組態更新的彈性，盡可能避免整個叢集停止和啟動。新增組態參數 Scheduling / SlurmSettings / QueueUpdateStrategy，以設定運算節點需要組態更新和取代時使用的偏好策略。 • 在 Amazon EC2 執行個體遇到容量不足問題時，透過可用的運算資源改善容錯移轉機制。當節點啟動因容量不足而失敗時，依可設定的時間量停用運算節點。 	<p>2022 年 7 月 27 日</p>

變更	描述	Date
	<ul style="list-style-type: none"> • 新增支援以掛載現有的 FSx for ONTAP 和 FSx for OpenZFS 檔案系統。 • 新增支援，以掛載現有 Amazon Elastic File Systems、FSx for Lustre、FSx for ONTAP 和 FSx for OpenZFS 檔案系統的多個執行個體。 • 在建立新的檔案系統時，新增對 FSx for Lustre Persistent_2 部署類型的 支援。 • 提示使用者在使用 叢集設定精靈 時，為支援的執行個體類型啟用 EFA。 • 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，以同時考慮手動關閉節點電源。 • 在產品 AMIs 中安裝 NVIDIA GDRCopy 2.3，以啟用低延遲 GPU 記憶體複製。 <p>變更：</p> <ul style="list-style-type: none"> • 將 EFA 安裝程式升級至 1.17.2 版。 <ul style="list-style-type: none"> • EFA 驅動程式： efa-1.16.0-1 • EFA 組態：efa-config-1.10-1 	

變更	描述	Date
	<ul style="list-style-type: none"> • EFA 設定檔：efa-profile-1.5-1 • Libfabric：libfabric-aws-1.16.0~amzn2.0-1 • RDMA 核心：rdma-core-41.0-2 • 開啟 MPI：openmpi40-aws-4.1.4-2 • 將 NICE DCV 升級至 2022.0-12760 版。 • 將 NVIDIA 驅動程式升級至 470.129.06 版。 • 將 NVIDIA Fabric Manager 升級到 470.129.06 版。 • 在根磁碟區和其他磁碟區中，將預設 EBS 磁碟區類型從 gp2 變更為 gp3。 • FSx for Lustre 檔案系統變更建立者 AWS ParallelCluster： <ul style="list-style-type: none"> • 將預設部署類型變更為 Scratch_2。 • 將 Lustre 伺服器版本變更為 2.12。 • 傳遞現有的 / true 時，不需要將 PlacementGroup PlacementGroup / Enabled 設定為 Id。 • 當 PlacementGroup / Enabled 明確設為 Id 時， 	

變更	描述	Date
	<p>不允許設定 Placement Group /false。</p> <ul style="list-style-type: none"> 將parallelcluster:cluster-name 標籤新增至 建立的所有資源 AWS ParallelCluster。 lambda:UntagResource 將 lambda:ListTags 和 新增至 AWS ParallelCluster API ParallelClusterUserRole 堆疊用於叢集更新。 Secured 啟用組態參數 / HeadNode / 時，僅限對IMDS根和叢集管理員使用者的 IPv6 Imds 存取。 使用自訂 AMI 時，請使用 AMI 根磁碟區大小，而非 ParallelCluster 預設值 35 GiB。您可以在叢集組態檔案中變更 值。 當組態參數 Scheduling / SlurmQueues // SpotPrice 低於所需的 Spot ComputeResources 請求履行價格下限時，自動停用運算機群。 在更新期間新增或移除區段時，在變更集中顯示 requested_value 和 current_value 值。 	

變更	描述	Date
	<ul style="list-style-type: none"> • 在深度學習 AMI 中停用 <code>aws-ubuntu-eni-helper</code> 服務，以避免在使用多個網路卡設定執行個體 <code>configure_nw_interface.sh</code> 時與衝突。AMIs • 移除對 Python 3.6 的支援。 • 使用多個網路卡設定執行個體時，將所有網路介面的 MTU 設定為 9001。 • 設定運算節點 FQDN 時，移除結尾的點。 • 在 <code>中</code> 管理靜態節點 <code>POWERING_DOWN</code> 。 • 不會在 <code>中</code> 取代動態節點，<code>POWER_DOWN</code> 因為任務可能仍在執行中。 • 只有在叢集組態中更新 <code>Scheduling</code> 參數時，才能在叢集更新時間重新啟動 <code>slurmctld</code> <code>clustermgtd</code> 和精靈。 • 更新 <code>slurmctld</code> <code>slurmdsystemd</code> 和服務檔案。 • <code>Secured</code> 啟用組態參數 <code>//</code> 時，將 IMDS <code>Imds</code> 的 IPv6 <code>HeadNode</code> 存取限制為僅限根和叢集管理員使用者。 • 設定 Slurm 組態 <code>AuthInfo=cred_expire=70</code> ，以減少節點無法使用時，重新排 	

變更	描述	Date
	<p>入佇列任務必須等待的時間，然後再重新開始。</p> <ul style="list-style-type: none"> • 升級第三方技術指南相依性： <ul style="list-style-type: none"> • apt-7.4.2 (來自 apt-7.4.0) • line-4.5.2 (從 line-4.0.1) • openssh-2.10.3 (從 openssh-2.9.1 開始) • pyenv-3.5.1 (來自 pyenv-3.4.2) • selinux-6.0.4 (來自 selinux-3.1.1) • yum-7.4.0 (來自 yum-6.1.1) • yum-epel-4.5.0 (來自 yum-epel-4.1.2) <p>錯誤修正：</p> <ul style="list-style-type: none"> • 修正預設行為，以便在建置自訂 AMI 時略過 AWS ParallelCluster 驗證和測試步驟。 • 修正中的檔案控制代碼洩漏 <code>computemgtd</code>。 • 修正偶爾導致啟動的執行個體因為尚未在 EC2 <code>DescribeInstances</code> 回應中提供而立即終止的競爭條件。 • 修正對使用 Arm 處理器之執行個體類型的 <code>DisableSi</code> 	

變更	描述	Date
	<p>multaneousMultithreading 參數支援。</p> <ul style="list-style-type: none">• 從先前版本升級時修正 AWS ParallelCluster API 堆疊更新失敗。在中新增用於ListImagePipelineImages 動作的資源模式EcrImageDeletionLambdaRole 。• 修正 AWS ParallelCluster API 在建立 FSx for Lustre 檔案系統時，新增從 Amazon S3 匯入或匯出所需的缺少許可。 <p>如需變更的詳細資訊，請參閱 GitHub 上的 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 今年至今僅提供文件的更新	<p>AWS ParallelCluster 僅限文件的更新。</p> <p>新區段：</p> <ul style="list-style-type: none"> • 最佳實務：預算提醒 V3 • 最佳實務：將叢集移至新的 AWS ParallelCluster 次要或修補程式版本 V3 • 使用 Amazon S3 V3 • 使用 Spot 執行個體 V3 • Slurm 叢集保護模式 V3 • AWS ParallelCluster 資源和標記 V3 • Amazon CloudWatch 儀表板 V3 • 與 Amazon CloudWatch Logs V3 整合 • Elastic Fabric Adapter V3 • AWS ParallelCluster AMI 自訂 V3 • 使用隨需容量保留 (ODCR) V3 啟動執行個體 V3 • AMI 修補和 Amazon EC2 執行個體替換 V3 • V3 AWS ParallelCluster 的運作方式 • 使用 AWS KMS 金鑰 V3 設定共用儲存加密 V3 • 在多個佇列模式叢集 V3 中執行任務 V3 	2022 年 7 月 6 日

變更	描述	Date
	<ul style="list-style-type: none">• 使用 AWS ParallelCluster API V3 <p>區段更新：</p> <ul style="list-style-type: none">• 最佳實務：網路效能 V3：新增使用 Elastic Fabric 轉接器的最佳實務。• V3 AWS 中的 Identity and Access Management 許可 AWS ParallelCluster：使用 AWS ParallelCluster Amazon FSx for Lustre 時，各種更新並新增其他叢集使用者政策。• V3 AWS ParallelCluster 故障診斷：各種更新。	

變更	描述	Date
AWS ParallelCluster 3.1.4 版已發行	<p>AWS ParallelCluster 3.1.4 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 如果秘密不存在，請新增 DirectoryService / PasswordSecretArn 的驗證以失敗。 <p>新增啟用 JWT 身分驗證 Slurm 的支援。</p> <p>變更：</p> <ul style="list-style-type: none">• 升級 Slurm 至 21.08.8-2 版。• 使用 JWT 支援建置 Slurm。• 傳遞現有的 / true 時，不需要將 PlacementGroup PlacementGroup / Enabled 設定為 Id。• <code>lambda:TagResource</code> 將新增至 <code>ParallelClusterUserRole</code> ParallelCluster API 堆疊用於建立叢集和建立映像。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 修正使用 <code>export-cluster-logs</code> 命令搭配 <code>--filters</code> 選項時匯出叢集日誌的功能。	2022 年 5 月 16 日

變更	描述	Date
	<ul style="list-style-type: none">• 修正 AWS Batch Docker 進入點，以使用/home共用目錄來協調Multi-node-Parallel任務執行。• 將 Slurm 運作狀態不佳的靜態節點設定為關閉時重設節點地址，以避免將容量不足的靜態節點視為引導失敗節點。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 3.1.3 版已發行	<p>AWS ParallelCluster 3.1.3 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 執行 SSH 金鑰建立以及 HOME 目錄建立，例如，在 SSH 登入期間、切換到另一個使用者時，以及以另一個使用者身分執行命令時。• 在組態參數 DirectoryService / DomainName 中新增對 FQDN 和 LDAP 辨別名稱的支援。新的驗證程式現在會檢查這兩個語法。• 部署在前端節點上的新 <code>update_directory_service_password.sh</code> 指令碼支援在 SSSD 組態中手動更新 Active Directory 密碼。密碼由 AWS Secrets Manager 從叢集組態擷取為。• 新增支援以在沒有預設 VPC 的環境中部署 API 基礎設施。 <p>變更：</p> <ul style="list-style-type: none">• 在透過 <code>build-image</code> 命令建立的 x86_64 官方 AMIs 和 AMIs 中停用更深層的 C-States，以確保高效能和低延遲。	2022 年 4 月 20 日

變更	描述	Date
	<ul style="list-style-type: none">• 作業系統套件更新和安全性修正。• 將 Amazon Linux 2 基礎映像變更為搭配核心 5.10 使用 AMIs。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 由於新的 EC2 Image Builder 政策，在映像建置成功DELETE_FAILED 之後修正中的建置映像堆疊。• 當包含多個網域地址時，修正組態參數 Directory Service/DomainAddr 轉換為 ldap_uri SSSD 屬性。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster 和 aws-parallelcluster-cookbook 套件CHANGELOG 的檔案。</p>	

變更	描述	Date
AWS ParallelCluster 3.1.2 版已發行	<p>AWS ParallelCluster 3.1.2 版已發行。</p> <p>變更：</p> <ul style="list-style-type: none">• 升級 Slurm 至版本 21.08.6 (從 21.08.5)。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 在沒有網際網路存取子網路中部署叢集時，修正運算節點上的 <code>/etc/hosts</code> 檔案更新。• 修正運算節點引導，在加入叢集之前等待暫時性磁碟機初始化。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster 套件CHANGELOG 的檔案。</p>	2022 年 3 月 2 日

變更	描述	Date
AWS ParallelCluster 3.1.1 版已發行	<p data-bbox="592 226 1003 306">AWS ParallelCluster 3.1.1 版已發行。</p> <ul data-bbox="592 352 1026 1780" style="list-style-type: none"><li data-bbox="592 352 1026 579">• 透過與透過 Directory AWS Service 管理的 Active Directory (AD) 網域整合，新增對多個使用者叢集環境的支援。<li data-bbox="592 600 1026 877">• 在叢集組態檔案中新增對 UseEc2Hostnames 的支援。設為 true 時，針對運算節點使用 Amazon EC2 預設主機名稱（例如 ip-1-2-3-4）。<li data-bbox="592 898 1026 978">• 新增在 沒有網際網路存取的子網路 中建立叢集的支援。<li data-bbox="592 999 1026 1079">• 為每個佇列新增對多個運算執行個體類型的支援。<li data-bbox="592 1100 1026 1234">• 新增使用 NVIDIA 卡在 ARM 執行個體上使用 Slurm 進行 GPU 排程的支援。<li data-bbox="592 1255 1026 1579">• 將 <code>cluster-name (-n)</code>、<code>region (-r)</code>、<code>image-id (-i)</code> 和 <code>/cluster-configuration image-configuration (-c)</code> 的縮寫旗標新增至 AWS ParallelCluster CLI。<li data-bbox="592 1600 1026 1780">• 新增對 FSx for Lustre AutoImportPolicy 參數 <code>NEW_CHANGED_DELETE</code> 選項的支援。	2022 年 2 月 10 日

變更	描述	Date
	<ul style="list-style-type: none"> • 將parallelcluster:compute-resource-name 標籤新增至運算節點使用的 EC2 LaunchTemplates 資源。 • 改善叢集內建立的安全群組，以便在為某些前端節點和/或佇列指定SecurityGroups 參數時，允許來自自訂安全群組的傳入連線。 • 安裝適用於 ARM 的 NVIDIA 驅動程式和 CUDA 程式庫。 <p>變更：</p> <ul style="list-style-type: none"> • 升級 Slurm 至版本 21.08.5 (從 20.11.8)。 • 將 Slurm 外掛程式升級至版本 21.08 (從 20.11)。 • 將 NICE DCV 升級到版本 2021.3-11591 (從 2021.1-10851)。 • 將 NVIDIA 驅動程式升級至版本 470.103.01 (從 470.57.02)。 • 將 NVIDIA Fabric Manager 升級到版本 470.103.01 (從 470.57.02)。 • 將 CUDA 升級到版本 11.4.4 (從 11.4.0)。 • Intel MPI 已更新至版本 2021 Update 4 (更新自版本 2019 Update 8)。如需詳 	

變更	描述	Date
	<p>細資訊，請參閱 Intel® MPI Library 2021 Update 4。</p> <ul style="list-style-type: none">• 將 PMIx 升級到版本 3.2.3 (從 3.1.5)。• 將失敗的運算節點傾印移除至 /home/logs/compute。運算節點日誌檔案可在 CloudWatch 和 Amazon EC2 主控台日誌中使用。• 啟用潛在隱藏 SlurmQueues 和 ComputeResources 長度驗證程式。• 在 Amazon Linux 2 上的執行個體啟動時間停用套件更新。• 在建置 AWS ParallelCluster 自訂映像時停用 Amazon EC2 ImageBuilder 增強型映像中繼資料。• 將 cloud-init 資料來源明確設定為 EC2。這可節省 Ubuntu 和 CentOS 平台的開機時間。• 在運算機群啟動範本名稱中使用運算資源名稱，而非執行個體類型。• 將 stderr 和 stdout 重新導向至 CLI 日誌檔案，以防止 pcluster CLI 輸出中出現不需要的文字。• 移動設定/安裝配方來分隔從主要技術指南呼叫的技術指	

變更	描述	Date
	<p>南。現有的進入點會維護並回溯相容。</p> <ul style="list-style-type: none"> • 在 AMI 建置期間下載 Intel HPC 平台的相依性，以避免在叢集建立期間接觸網際網路。 • 設定 Slurm 節點時，請勿-從運算資源名稱中分割。 • 未安裝 NVIDIA 驅動程式時，請勿在 Slurm 中設定 GPUs。 • 修正中的ecs:ListContainerInstances 許可BatchUserRole 。 • 修正未指定字首、先前匯出至字None首的叢集日誌匯出。 • 在叢集更新失敗時，不會執行修正轉返。 • 修正中的ecs:ListContainerInstances 許可BatchUserRole 。 • 如果KmsKeyId指定不支援的 RootVolume ，HeadNode請引發錯誤來修正 的結構描述。 • 修正要在 CloudWatch Dashboard 中顯示的 Amazon FSx 遺失指標。 • 修正 EfaSecurityGroupValidator 。 <p>先前，當提供自訂安全群組</p>	

變更	描述	Date
	<p>並啟用 EFA 時，它可能會導致錯誤失敗。</p> <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。</p>	
AWS ParallelCluster 3.0.3 版已發行	<p>AWS ParallelCluster 3.0.3 版已發行。</p> <ul style="list-style-type: none">在 Amazon Linux 2 上停用 <code>log4j-cve-2021-44228-hotpatch</code> 代理程式 (Log4jHotPatch)，以避免潛在的效能降低。如需詳細資訊，請參閱 Apache Log4j 的 Amazon Linux 熱修補程式公告。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster 和 aws-parallelcluster-cookbook 套件CHANGELOG 的檔案。</p>	2022 年 1 月 17 日

變更	描述	Date
AWS ParallelCluster 3.0.2 版已發行	<p data-bbox="592 226 1003 306">AWS ParallelCluster 3.0.2 版已發行。</p> <p data-bbox="592 352 1016 432">將 Elastic Fabric Adapter 安裝程式升級至 1.14.1</p> <ul data-bbox="592 478 1016 1335" style="list-style-type: none">• EFA 組態：efa-config-1.9-1 (來自 efa-config-1.9)• EFA 設定檔：efa-profile-1.5-1 (來自 efa-profile-1.5)• EFA 核心模組：efa-1.14.2 (來自 efa-1.13.0)• RDMA 核心：rdma-core-37.0 (來自 rdma-core-35)• Libfabric：libfabric-1.13.2 (來自 libfabric-1.13.0)• 開啟 MPI：openmpi40-aws-4.1.1-2 (無變更) <p data-bbox="592 1407 1016 1587">如果執行個體類型支援，一律會啟用 GPUDirect RDMA。GdrSupport 組態選項沒有效果。</p> <p data-bbox="592 1633 984 1810">如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-parallelcluster-cookbook 和 aws-</p>	2021 年 11 月 5 日

變更	描述	Date
	parallelcluster-node 套件CHANGELOG 的檔案。	

變更	描述	Date
AWS ParallelCluster 3.0.1 版已發行	<p>AWS ParallelCluster 3.0.1 版已發行。</p> <p>叢集組態遷移工具</p> <ul style="list-style-type: none">• 客戶現在可以將叢集組態從第 2 AWS ParallelCluster 版格式遷移到以 YAML 為基礎的第 3 AWS ParallelCluster 版格式。如需詳細資訊，請參閱 pcluster3-config-converter。 <p>前端節點可以停止</p> <ul style="list-style-type: none">• 停止運算機群後，可以使用 Amazon EC2 主控台或 stop-instances AWS CLI 命令來停止前端節點，並在稍後重新啟動。 <p>從 ~/.aws/config 檔案讀取的預設 AWS 區域</p> <ul style="list-style-type: none">• 對於 pcluster 命令，如果未在組態檔案、環境或命令列中指定 AWS 區域，則會使用 ~/.aws/config 檔案 [default] 區段中 region 設定中指定的預設 AWS 區域。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-paral</p>	2021 年 10 月 27 日

變更	描述	Date
	elcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。	

變更	描述	Date
AWS ParallelCluster 3.0.0 版已發行	<p data-bbox="592 226 1003 306">AWS ParallelCluster 3.0.0 版已發行。</p> <p data-bbox="592 352 1019 432">透過 Amazon API Gateway 支援叢集管理</p> <ul data-bbox="592 478 1019 705" style="list-style-type: none">• 客戶現在可以使用 Amazon API Gateway 透過 HTTP 端點管理和部署叢集。這會為指令碼或事件驅動的工作流程開啟新的可能性。 <p data-bbox="623 751 1019 1020">AWS ParallelCluster 命令列界面 (CLI) 也已重新設計，以與此 API 相容，並包含新的 JSON 輸出選項。這項新功能可讓客戶使用 CLI 實作類似的建置區塊功能。</p> <p data-bbox="592 1100 857 1134">改善自訂 AMI 建立</p> <ul data-bbox="592 1180 1026 1642" style="list-style-type: none">• 客戶現在可以存取更強大的程序，以使用 EC2 Image Builder 建立和管理自訂 AMIs。自訂 AMIs 現在可以透過單獨的 AWS ParallelCluster 組態檔案進行管理，並且可以使用 AWS ParallelCluster 命令列界面中的 pcluster build-image 命令來建立。 <p data-bbox="592 1717 961 1852">如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster、aws-paral</p>	2021 年 9 月 10 日

變更	描述	Date
	lcluster-cookbook 和 aws-parallelcluster-node 套件CHANGELOG 的檔案。	

PCUI

變更	描述	Date
PCUI 2025.04.0 版已發行	<p>PCUI 2025.04.0 版已發行</p> <p>重大變更：</p> <ul style="list-style-type: none"> 移除 PC 版本的預設值。現在，使用者必須指定要使用的 PC 版本。 <p>功能：</p> <ul style="list-style-type: none"> 新增新的堆疊參數 'AdditionalPolicies PCAPI'，以新增 ParallelCluster API Lambda 角色的自訂許可，以及預設的許可。 <p>錯誤修正：</p> <ul style="list-style-type: none"> 讓 PCUI 範本使用並傳回正確的 URLs，以修正私有子網路中的 PCUI 部署。 修正在任務狀態索引標籤中防止載入超過 200 個任務的問題。(請參閱 https://github.com/aws/aws-parallelcluster-ui/issues/376 : //)。 	2025 年 4 月 16 日

變更	描述	Date
	<p>安全性：</p> <ul style="list-style-type: none">• 從 3.9 升級至 3.12。• 從 7.0.3 升級到 7.0.6，以解決漏洞 CVE-2024-21538。• 從 2.31.0 升級到 2.32.0 以解決 CVE-2024-35195。• 將 urllib3 從 1.26.18 升級到 1.26.19，以解決 CVE-2024-37891。• 從 42.0.4 升級到 44.0.1 以解決 CWE-1395。• 從 2023.7.22 升級到 2024.7.4 以解決 CVE-2024-39689。• 從 3.1.3 升級到 3.1.6 以解決 CVE-2024-56201 和 CVE-2024-56326。• 將 serverless_wsgi.py 升級至 3.0.5 版。• 從 2.3.8 升級至 3.0.6 版，以解決 CVE-2024-34069、CVE-2024-49766 和 CVE-2024-49767。• 從 1.6.7 升級到 1.8.2 版，以解決 CVE-2024-39338。• 從 14.1.1 升級 Next.js 到 14.2.25 版，以解決 CVE-2024-51479、CVE-2024-46982 和 CVE-2025-29927。• 從 3.4 升級至 3.7 版以解決 CVE-2024-3651。	

變更	描述	Date
	<ul style="list-style-type: none">從 3.3.7 升級至 3.3.8 版以解決 CVE-2024-55565。將 python-jose 從 3.3.0 升級至 3.4.0 版，以解決 CVE-2022-29217。	
PCUI 2024.11.0 版已發行	PCUI 2024.11.0 版已發行 錯誤修正： <ul style="list-style-type: none">明確設定 ECR 私有儲存庫的政策，以防止在會影響 Lambda 函數的堆疊更新上移除政策。此政策包含 Lambda 函數擷取程式碼所需的許可。	2024 年 11 月 22 日

變更	描述	Date
PCUI 2024.10.0 版已發行	<p data-bbox="592 226 954 260">PCUI 2024.10.0 版已發行</p> <p data-bbox="592 306 678 340">變更：</p> <ul data-bbox="592 386 1019 877" style="list-style-type: none"><li data-bbox="592 386 1019 466">• 新增對 AWS ParallelCluster 3.11.1 的支援。<li data-bbox="592 491 1019 571">• 在精靈中新增對隨需容量預留和容量區塊的支援。<li data-bbox="592 596 1019 726">• 將 g6、m7 和 p5 系列新增至精靈中支援的執行個體類型清單。<li data-bbox="592 751 1019 877">• 新增堆疊選用參數，以設定 PCUI 和 Cognito 的自訂網域。 <p data-bbox="592 953 743 987">錯誤修正：</p> <ul data-bbox="592 1033 1019 1113" style="list-style-type: none"><li data-bbox="592 1033 1019 1113">• 修正中斷自訂網域設定的錯誤。 <p data-bbox="592 1188 711 1222">安全性：</p> <ul data-bbox="592 1268 1029 1566" style="list-style-type: none"><li data-bbox="592 1268 1029 1398">• 從 3.0.10 升級 Flask-CORS 至 4.0.2，以解決漏洞 CVE-2024-6221。<li data-bbox="592 1423 1029 1503">• 從 13.0.3 升級至 15.2.5 以解決漏洞 CVE-2024-4068。<li data-bbox="592 1528 818 1566">• 完整變更日誌	2024 年 10 月 22 日

變更	描述	Date
PCUI 2024.05.0 版已發行	<p>PCUI 2024.05.0 版已發行。</p> <p>錯誤修正：</p> <ul style="list-style-type: none">修正當使用者開啟任務狀態面板時，前端封鎖 UI 的錯誤。完整變更日誌	2024 年 5 月 14 日
PCUI 2024.04.0 版已發行	<p>PCUI 2024.04.0 版已發行。</p> <p>功能：</p> <ul style="list-style-type: none">新增對 3.9.1 AWS ParallelCluster 版的支援完整變更日誌	2024 年 4 月 17 日

變更	描述	Date
PCUI 2024.03.0 版已發行	<p>PCUI 2024.03.0 版已發行。</p> <p>功能：</p> <ul style="list-style-type: none"> • 新增對 3.9.0 AWS ParallelCluster 版的支援 • 新增對 Ubuntu 22.04 和 Red Hat Enterprise Linux 9 的支援 • 已取代的 Ubuntu 18.04 <p>錯誤修正</p> <ul style="list-style-type: none"> • 已修正使用許多叢集時，導致某些叢集未出現的問題 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	2024 年 3 月 12 日
PCUI 2024.02.0 版已發行	<p>PCUI 2024.02.0 版已發行</p> <p>變更：</p> <ul style="list-style-type: none"> • 將 Lambda 執行期環境更新為 Python v3.9 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	2024 年 2 月 8 日

變更	描述	Date
PCUI 2023.12.0 版已發行	<p>PCUI 2023.12.0 版已發行。</p> <p>功能：</p> <ul style="list-style-type: none">• 新增使用私有聯網進行 PCUI 部署的支援。• 新增可選擇將許可界限套用至 PCUI 和 PCAPI 基礎設施建立的每個 IAM 角色的可能性• 新增可選擇將字首套用至 PCUI 和 PCAPI 基礎設施建立的每個 IAM 角色和政策的的可能性。• 新增對 ParallelCluster 3.8.0 版的支援，在精靈中沒有功能同位。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	2023 年 12 月 21 日

變更	描述	Date
PCUI 2023.10.0 版已發行	<p data-bbox="592 226 971 260">PCUI 2023.10.0 版已發行。</p> <p data-bbox="592 306 678 340">功能：</p> <ul data-bbox="592 386 1019 613" style="list-style-type: none">• 新增對 ParallelCluster 3.7.2 的支援，精靈中的功能同位僅限於 FSx 檔案快取，以及記憶體型排程與多個執行個體類型的相容性。 <p data-bbox="592 688 743 722">錯誤修正：</p> <ul data-bbox="592 768 1019 898" style="list-style-type: none">• 已修正當 PCUI 沒有與 Cost Explorer 互動的許可時，導致 UI 錯誤的問題。 <p data-bbox="592 974 657 1008">改進</p> <ul data-bbox="592 1054 1019 1184" style="list-style-type: none">• 透過將存取權杖 TTL 從 10 分鐘減少為 5 分鐘來改善安全性。 <p data-bbox="592 1260 1019 1390">如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	2023 年 10 月 20 日

變更	描述	Date
PCUI 2023.06.0 版已發行	<p>PCUI 2023.06.0 版已發行。</p> <p>變更：</p> <ul style="list-style-type: none">• 已將預設 AWS ParallelCluster API 版本升級至 3.6.0。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 已修正 AWS GovCloud (美國西部) 區域的中斷部署。• 分割面板現在可在建立開始後正確載入叢集詳細資訊。 <p>備註：</p> <ul style="list-style-type: none">• 成本監控功能不適用於 AWS GovCloud (US) 區域。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	2023 年 6 月 7 日

變更	描述	Date
PCUI 2023.05.0 版已發行	<p>PCUI 2023.05.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 從 3.6.0 AWS ParallelCluster 版開始，新增對 RHEL 8 的支援。• 新增叢集成本監控。• 從 3.6.0 AWS ParallelCluster 版開始，增加佇列和運算資源配額。 <p>變更：</p> <ul style="list-style-type: none">• 已改善叢集建立精靈使用者介面。• 增加 PCUI 部署的速度。• 已改善新增使用者的界面。• 根據預設，佇列位於前端節點子網路中。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 在叢集建立完成後切換到正確的區域。• 修正「編輯叢集」功能中的載入指標顯示。• 修正移除 EBS SnapshotId 屬性時的叢集建立。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件CHANGELOG 的檔案。</p>	2023 年 5 月 16 日

變更	描述	Date
PCUI 2023.04.0 版已發行	<p>PCUI 2023.04.0 版已發行。</p> <p>增強功能：</p> <ul style="list-style-type: none">• 叢集建立精靈重新設計。• 叢集日誌頁面重新設計。• 新增共用儲存體的自訂名稱設定。• 在將儲存新增至叢集時新增多個儲存體選擇。• 新增對 Amazon EFS 和 FSx for Lustre 的 DeletionPolicy 支援。• 在叢集組態中新增 ImdsSupport 設定。• 新增對 C7 執行個體類型的支援。• 新增教學 還原至先前的 AWS Systems Manager 文件版本。 <p>變更：</p> <ul style="list-style-type: none">• 叢集組態 YAML 大小上限為 1MB。• 由於使用 Boto3 IAM 臨時憑證的授權，使用者不會登出。• 選取 HPC 執行個體時停用多執行緒選項。• 已移除叢集建立頁面上的停用轉返。	2023 年 4 月 17 日

變更	描述	Date
	<ul style="list-style-type: none">• 在提供必要資訊之前，使用者無法使用 PCUI。• 最多可新增 10 個佇列。• 在 PCUI 安裝期間，不會覆寫 SSM-SessionManager RunShell 文件。 <p>錯誤修正：</p> <ul style="list-style-type: none">• 修正中斷的重設密碼連結。• 修正因 EcrPrivateRepository 不是空的 delete stack 而造成的損壞• 修正多個使用者管理屬性區段中產生 SSH 金鑰核取方塊的初始化問題。• 已修正當機原因為具有未定義屬性的任務。• 已修正 SCRATCH FSx 設定。• 已修正啟動和停止執行個體按鈕，在按一下後仍然啟用。 <p>如需變更的詳細資訊，請參閱 GitHub 上 aws-parallelcluster-ui 套件 CHANGELOG 的檔案。</p>	

Terraform

變更	描述	Date
Terraform Provider for AWS ParallelCluster 1.1.0 已發行	錯誤修正： <ul style="list-style-type: none">修正使用 ParallelCluster API 3.11.x 部署具有登入節點的叢集時，導致 terraform-apply 失敗的問題。	2024 年 12 月 6 日
發行適用於 AWS ParallelCluster 1.1.0 的 Terraform 模組	變更： <ul style="list-style-type: none">在所有模組範例中使用 AWS ParallelCluster Terraform Provider 1.x。在堆疊名稱為 ParallelCluster API 的所有範例中，使用 ParallelClusterAPI API 3.11.1。在所有模組範例中部署登入節點。	2024 年 12 月 6 日
Terraform Provider for AWS ParallelCluster 1.0.0 已發行	功能： <ul style="list-style-type: none">完整變更日誌	2024 年 6 月 26 日
發行適用於 AWS ParallelCluster 1.0.0 的 Terraform 模組	功能： <ul style="list-style-type: none">完整變更日誌	2024 年 6 月 26 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。