



使用者指南

AWS PCS



AWS PCS: 使用者指南

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 PCS ?	1
概念	1
開始使用 AWS PCS	3
先決條件	4
註冊 AWS 並建立管理使用者	5
安裝 AWS CLI 適用於 AWS PCS 的	6
所需的 IAM 許可	7
使用 CloudFormation	7
建立 VPC 和子網路	7
尋找叢集 VPC 的預設安全群組	9
建立安全群組	9
建立安全群組	9
建立 叢集	10
在 Amazon EFS 中建立共用儲存	11
在 FSx for Lustre 中建立共用儲存	12
建立運算節點群組	13
建立執行個體設定檔	13
建立啟動範本	15
建立登入節點的運算節點群組	16
建立任務的運算節點群組	17
建立佇列	18
連線至您的叢集	19
探索叢集環境	20
變更使用者	20
使用共用檔案系統	20
與 Slurm 互動	20
執行單一節點任務	21
使用 Slurm 執行多節點 MPI 任務	23
刪除您的 AWS 資源	26
CloudFormation 和 AWS PCS 入門	29
使用 CloudFormation 建立叢集	29
連接至叢集	31
清除叢集	31
AWS PCS CloudFormation 範本的一部分	32

標頭	32
中繼資料	33
Parameters	33
映射項目	35
Resources	35
輸出	39
建立範例叢集的範本	40
叢集	42
建立叢集	42
先決條件	43
建立 AWS PCS 叢集	43
更新叢集	46
叢集更新的優點	46
支援的組態變更	47
限制	47
叢集更新先決條件	47
更新程序和任務影響	47
在更新期間計費	48
更新叢集	48
常見問答集	50
故障診斷	51
刪除叢集	52
刪除 AWS PCS 叢集時的考量事項	52
刪除叢集	52
叢集大小	53
叢集秘密	54
使用 AWS Secrets Manager 尋找叢集秘密	55
使用 AWS PCS 尋找叢集秘密	55
取得 Slurm 叢集秘密	56
秘密輪換	57
運算節點群組	62
建立運算節點群組	62
先決條件	63
在 AWS PCS 中建立運算節點群組	63
更新運算節點群組	67
更新 AWS PCS 運算節點群組的選項	68

更新 AWS PCS 運算節點群組時的考量事項	68
更新 AWS PCS 運算節點群組	69
刪除運算節點群組	70
刪除運算節點群組時的考量事項	71
刪除運算節點群組	71
取得運算節點群組詳細資訊	72
尋找運算節點群組執行個體	75
使用啟動範本	77
概觀	77
建立基本的啟動範本	78
使用 Amazon EC2 使用者資料	80
範例：從套件儲存庫安裝軟體	82
範例：從 S3 儲存貯體執行指令碼	82
範例：設定全域環境變數	84
範例：使用 EFS 檔案系統做為共用主目錄	84
Capacity Reservations	86
搭配 AWS PCS 使用 ODCRs	86
容量區塊	88
有用的啟動範本參數	93
開啟詳細的 CloudWatch 監控	93
執行個體中繼資料服務第 2 版 (IMDS v2)	94
佇列	95
建立佇列	95
先決條件	95
在 AWS PCS 中建立佇列	95
更新佇列	97
更新 AWS PCS 佇列時的考量事項	98
更新 AWS PCS 佇列	98
刪除佇列	99
刪除佇列時的考量事項	100
刪除佇列	100
登入節點	102
使用運算節點群組登入	102
建立登入節點的 AWS PCS 運算節點群組	102
更新登入節點的 AWS PCS 運算節點群組	103
刪除登入節點的 AWS PCS 運算節點群組	103

使用獨立執行個體做為登入節點	103
步驟 1 – 擷取目標 AWS PCS 叢集的地址和秘密	104
步驟 2 – 啟動 EC2 執行個體	105
步驟 3 – 在執行個體上安裝 Slurm	106
步驟 4 – 擷取和存放叢集秘密	106
步驟 5 – 設定 AWS PCS 叢集的連線	107
步驟 6 – (選用) 測試連線	108
將獨立登入節點連接到多個叢集	109
先決條件	110
指令碼程式碼	111
使用指令碼	119
聯網	122
VPC 和子網要求	122
VPC 要求和注意事項	122
子網需求和注意事項	123
建立 VPC	124
先決條件	125
建立 Amazon VPC	125
Security groups (安全群組)	128
安全群組要求	129
多個網路介面	130
置放群組	131
使用 Elastic Fabric Adapter (EFA)	132
識別啟用 EFA 的 EC2 執行個體	133
建立安全群組以支援 EFA 通訊	133
(選用) 建立置放群組	135
建立或更新 EC2 啟動範本	135
建立或更新 EFA 的運算節點群組	136
(選用) 測試 EFA	136
(選用) 使用 CloudFormation 範本建立啟用 EFA 的啟動範本	138
網路檔案系統	140
使用網路檔案系統的考量事項	140
網路掛載範例	140
Amazon Machine Images (AMI)	146
使用範例 AMIs	146
尋找目前的 AWS PCS 範例 AMIs	146

進一步了解 AWS PCS 範例 AMIs	148
建置與 AWS PCS 相容的自有 AMIs	148
自訂 AMI	148
步驟 1 – 啟動暫時執行個體	149
步驟 2 – 安裝 AWS PCS 代理程式	150
步驟 3 – 安裝 Slurm	152
步驟 4 – (選用) 安裝其他驅動程式、程式庫和應用程式軟體	155
步驟 5 – 建立與 AWS PCS 相容的 AMI	156
步驟 6 – 搭配 AWS PCS 運算節點群組使用自訂 AMI	156
步驟 7 – 終止暫時執行個體	158
建置 AMIs 安裝程式	158
AWS PCS 代理程式軟體安裝程式	159
Slurm 安裝程式	159
支援的作業系統	160
支援的執行個體類型	160
支援的 Slurm 版本	160
使用檢查總和驗證安裝程式	160
AMIs 的版本備註	167
x86_64 (AL2) AMIs	167
Arm64 (AL2) AMIs 範例	170
支援的作業系統	174
AWS PCS 代理程式版本	176
Slurm	179
Slurm 版本	179
AWS PCS 中支援的 Slurm 版本	179
AWS PCS 中不支援的 Slurm 版本	181
版本備註	181
常見問答集	183
Slurm 會計	185
修改會計設定	186
重要概念	186
取得現有 AWS PCS 叢集的會計組態	187
Slurm REST API	188
常用案例	188
要求與限制	188
啟用 REST API	189

REST API 身分驗證	191
使用 REST API	195
REST API 常見問答集	197
Slurm 重新啟動	199
Slurm 重新開機的優點	199
何時使用 Slurm 重新開機	200
限制	200
重新啟動運算節點	200
取消重新啟動	202
常見問答集	202
疑難排解	204
自訂 Slurm 設定	205
自訂 Slurm 設定的優點	205
設定自訂設定	205
驗證和錯誤處理	206
限制	206
叢集設定	207
運算節點群組設定	209
佇列設定	209
疑難排解	210
自訂 Cgroup 設定	211
設定 cgroup 設定	211
叢集支援的 cgroup 設定	212
自訂 SlurmDBD 設定	212
設定 slurmdbd 設定	212
叢集支援的 slurmdbd 設定	213
SPANK 外掛程式	214
安裝 SPANK 外掛程式	214
設定 SPANK 外掛程式	215
SPANK 外掛程式常見問答集	216
Slurm CLI 篩選條件外掛程式	216
要求	216
限制和安全考量	217
設定 CLI 篩選條件外掛程式	217
使用 Amazon S3 部署 CLI 篩選條件外掛程式指令碼	221
翻譯任務提交外掛程式指令碼	222

常見問答集	223
疑難排解	224
安全	227
資料保護	227
靜態加密	228
傳輸中加密	229
金鑰管理	229
網際網路流量隱私權	229
加密 API 流量	230
加密資料流量	230
加密 EBS 磁碟區的 KMS 金鑰政策	230
VPC 介面端點 (AWS PrivateLink)	236
考量事項	236
建立介面端點	236
建立端點政策	236
身分和存取權管理	237
目標對象	238
使用身分驗證	238
使用政策管理存取權	239
AWS Parallel Computing Service 如何與 IAM 搭配使用	241
身分型政策範例	245
AWS 受管政策	248
服務連結角色	250
EC2 Spot 角色	251
最低許可	252
執行個體設定檔	258
疑難排解	263
法規遵循驗證	264
恢復能力	264
基礎設施安全性	265
漏洞分析和管理的	265
預防跨服務混淆代理人	266
作為運算節點群組一部分佈建之 Amazon EC2 執行個體之 IAM 角色	267
安全最佳實務	268
AMI 相關安全性	268
Slurm Workload Manager 安全性	268

監控和記錄	269
網路安全	269
日誌記錄和監控	270
任務完成日誌	270
先決條件	271
設定任務完成日誌	272
如何尋找任務完成日誌	273
任務完成日誌欄位	274
任務完成日誌範例	277
排程器日誌	281
先決條件	281
設定排程器日誌	282
排程器日誌串流路徑和名稱	283
排程器日誌記錄範例	284
使用 CloudWatch 進行監控	285
監控指標	285
監控執行個體	286
CloudTrail 日誌	294
AWS CloudTrail 中的 PCS 資訊	294
從 AWS PCS 了解 CloudTrail 日誌檔案項目	295
端點和服務配額	297
服務端點	297
Service Quotas	300
內部配額	301
其他服務的相關配額 AWS	301
疑難排解	303
EC2 執行個體會重新啟動後終止並取代	303
對 AWS PCS 中的運算節點引導和註冊問題進行故障診斷	304
Slurm 如何在 AWS PCS 上運作	305
擷取執行個體日誌	305
從執行個體 ID 擷取 VPC/Subnet/Security 群組	306
節點註冊問題	307
Slurm 叢集聯結問題	309
任務提交 MaxJobCount 限制	312
文件歷史紀錄	313
AWS 詞彙表	329

..... CCCXX

什麼是 AWS 平行運算服務？

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

主題

- [AWS PCS 中的概念](#)

AWS PCS 中的概念

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

叢集

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

運算節點群組

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

佇列

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

系統管理員

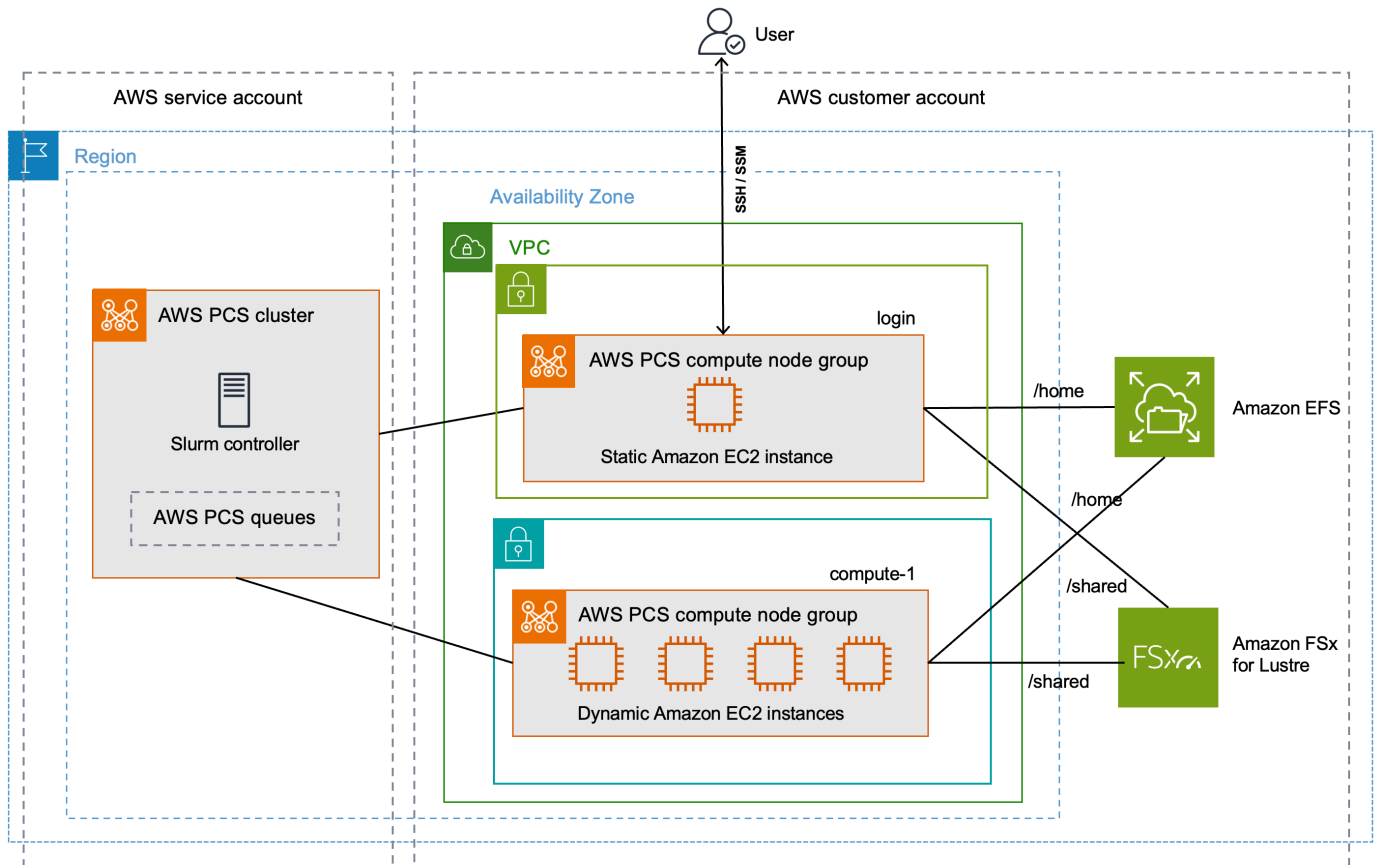
系統管理員會部署、維護和操作叢集。他們可以透過 AWS 管理主控台、AWS PCS API 和 AWS SDK 存取 AWS PCS。他們可以透過 SSH 或 存取特定叢集 AWS Systems Manager，在其中執行管理任務、執行任務、管理資料，以及執行其他以 shell 為基礎的活動。如需詳細資訊，請參閱 [AWS Systems Manager 文件](#)。

最終使用者

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

平行 AWS 運算服務入門

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



教學課程叢集設計具有下列重要元件：

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

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

Note

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

主題

- [開始使用 AWS PCS 的先決條件](#)
- [使用 AWS CloudFormation 搭配 AWS PCS 教學課程](#)
- [建立 AWS PCS 的 VPC 和子網路](#)
- [建立 AWS PCS 的安全群組](#)
- [在 AWS PCS 中建立叢集](#)
- [在 Amazon Elastic File System 中為 AWS PCS 建立共用儲存](#)
- [在 Amazon FSx for Lustre 中為 AWS PCS 建立共用儲存](#)
- [在 AWS PCS 中建立運算節點群組](#)
- [建立佇列以管理 AWS PCS 中的任務](#)
- [連線至 AWS PCS 叢集](#)
- [探索 AWS PCS 中的叢集環境](#)
- [在 AWS PCS 中執行單一節點任務](#)
- [在 AWS PCS 中使用 Slurm 執行多節點 MPI 任務](#)
- [刪除 AWS PCS AWS 的資源](#)

開始使用 AWS PCS 的先決條件

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

主題

- [註冊 AWS 並建立管理使用者](#)

- [安裝 AWS CLI 適用於 AWS PCS 的](#)
- [AWS PCS 所需的 IAM 許可](#)

註冊 AWS 並建立管理使用者

完成下列任務以設定 AWS 平行運算服務 (AWS PCS)。

主題

- [註冊 AWS 帳戶](#)
- [建立具有管理存取權的使用者](#)

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 會在註冊程序完成後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](#)。

安裝 AWS CLI 適用於 AWS PCS 的

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

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

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

```
aws pcs help
```

AWS PCS 所需的 IAM 許可

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

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

使用 AWS CloudFormation 搭配 AWS PCS 教學課程

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

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

如需詳細資訊，請參閱[CloudFormation 和 AWS PCS 入門](#)。

建立 AWS PCS 的 VPC 和子網路

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

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

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

- 在提供堆疊名稱下：
- 在堆疊名稱下，輸入：

hpc-networking

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

10.3.0.0/16

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

10.3.0.0/20

- 在 CidrPrivateSubnetA 下，輸入：

10.3.128.0/20

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

10.3.16.0/20

- 在 CidrPrivateSubnetB 下，輸入：

10.3.144.0/20

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

10.3.32.0/20

- 在 CidrPrivateSubnetC 下，輸入：

10.3.160.0/20

- 在功能下：
 - 勾選核取方塊以確認 AWS CloudFormation 可能會建立 IAM 資源。

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

尋找叢集 VPC 的預設安全群組

若要在新的 VPC 中尋找預設安全群組的 ID，請遵循下列程序：

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

建立 AWS PCS 的安全群組

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

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

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

建立 AWS PCS 的安全群組

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

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

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

- 在提供堆疊名稱下
- 在堆疊名稱下，輸入：

```
getstarted-sg
```

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

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

已建立兩個安全群組，名稱為：

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

在 AWS PCS 中建立叢集

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

若要建立叢集

1. 開啟 [AWS PCS 主控台](#)，然後選擇建立叢集。
2. 在叢集詳細資訊區段中，輸入下列欄位：
 - 叢集名稱 – 輸入 get-started
 - 排程器 – 選取 Slurm 25.05 版
 - 控制器大小 – 選取小型
3. 在聯網區段中，選取下列欄位的值：

- VPC – 選擇名為的 VPC `hpc-networking:Large-Scale-HPC`
 - 子網路 – 選取名稱開頭為的子網路 `hpc-networking:PrivateSubnetA`
 - 安全群組 – 選取名為的叢集安全群組 `cluster-getstarted-sg`
4. 選擇 Create Cluster (建立叢集)。

Note

在佈建叢集時，狀態欄位會顯示建立。建立叢集可能需要幾分鐘的時間。

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

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

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

建立 Amazon EFS 檔案系統

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

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

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

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

建立 FSx for Lustre 檔案系統

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

Note

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

在 AWS PCS 中建立運算節點群組

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

主題

- [建立 AWS PCS 的執行個體描述檔](#)
- [建立 AWS PCS 的啟動範本](#)
- [在 AWS PCS 中建立登入節點的運算節點群組](#)
- [建立運算節點群組以在 AWS PCS 中執行運算任務](#)

建立 AWS PCS 的執行個體描述檔

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

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

建立角色和執行個體描述檔

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

JSON

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

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

建立 AWS PCS 的啟動範本

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

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

存取 CloudFormation 範本

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

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-1t-efs-fsx1.yaml
```

使用 CloudFormation 範本建立 EC2 啟動範本

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

- 在提供堆疊名稱下：
 - 在堆疊名稱下，輸入 `getstarted-1t`。
- 在參數下：
 - 在安全性下
 - 針對 `VpcSecurityGroupId`，選取叢集 VPC `default` 中名為 的安全群組。
 - 針對 `ClusterSecurityGroupId`，選取名為 的群組 `cluster-getstarted-sg`
 - 針對 `SshSecurityGroupId`，選取名為 的群組 `inbound-ssh-getstarted-sg`
 - 對於 `SshKeyName`，選取您偏好的 SSH 金鑰對。
 - 檔案系統下
 - 對於 `EfsFileSystemId`，輸入您在教學中稍早建立的 EFS 檔案系統中的檔案系統 ID。
 - 對於 `FSxLustreFileSystemId`，輸入您在教學中稍早建立的 FSx for Lustre 檔案系統的檔案系統 ID。
 - 針對 `FSxLustreFileSystemMountName`，輸入相同 FSx for Lustre 檔案系統的掛載名稱。
- 選擇下一步，然後再次選擇下一步。

- 選擇提交。

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

Note

若要查看 CloudFormation 範本建立的所有資源，請開啟 [CloudFormation 主控台](#)。選擇 getstarted-1t 堆疊，然後選擇 Resources (資源) 索引標籤。

在 AWS PCS 中建立登入節點的運算節點群組

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

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

建立運算節點群組

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

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

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

- 選擇建立運算節點群組。

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

建立運算節點群組以在 AWS PCS 中執行運算任務

在此步驟中，您將啟動可彈性擴展的運算節點群組，以執行提交至叢集的任務。

建立運算節點群組

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

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

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

- 選擇建立運算節點群組。

狀態欄位會顯示正在佈建運算節點群組時的建立。

⚠ Important

等待狀態欄位顯示作用中，然後再繼續本教學課程的下一個步驟。

建立佇列以管理 AWS PCS 中的任務

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

在此步驟中，您將建立佇列，使用運算節點群組來處理任務。

建立佇列

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

⚠ Important

`compute-1` 群組的狀態必須處於作用中狀態，才能繼續下一個步驟。

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

在建立佇列時，狀態欄位會顯示建立。

⚠ Important

等待狀態欄位顯示作用中，然後再繼續本教學課程的下一個步驟。

連線至 AWS PCS 叢集

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

連線至登入節點

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

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

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

Session Manager

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

SSH

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

Note

執行個體的使用者名稱 `ec2-user` 不是 `root`。

探索 AWS PCS 中的叢集環境

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

變更使用者

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

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

使用共用檔案系統

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

```
[ec2-user@ip-10-3-6-103 ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  3.8G         0  3.8G   0% /dev
tmpfs                      3.9G         0  3.9G   0% /dev/shm
tmpfs                      3.9G    556K  3.9G   1% /run
tmpfs                      3.9G         0  3.9G   0% /sys/fs/cgroup
/dev/nvme0n1p1             24G       18G   6.6G  73% /
127.0.0.1:/                8.0E         0  8.0E   0% /home
10.3.132.79@tcp:/z1shxbev  1.2T    7.5M  1.2T   1% /shared
tmpfs                      780M         0  780M   0% /run/user/0
tmpfs                      780M         0  780M   0% /run/user/1000
```

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

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

與 Slurm 互動

主題

- [列出佇列和節點](#)
- [顯示任務](#)

列出佇列和節點

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

```
[ec2-user@ip-10-3-6-103 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
demo      up    infinite    4  idle~ compute-1-[1-4]
[ec2-user@ip-10-3-6-103 ~]$
```

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

顯示任務

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

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

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

在 AWS PCS 中執行單一節點任務

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

連線至叢集的登入節點，並在其 shell 提示下執行下列命令。

- 成為預設使用者。變更為共用目錄。

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

- 使用下列命令來建立範例任務指令碼：

```
cat << EOF > job.sh
#!/bin/bash
#SBATCH -J single
#SBATCH -o single.%j.out
#SBATCH -e single.%j.err

echo "This is job \${SLURM_JOB_NAME} [\${SLURM_JOB_ID}] running on \
\${SLURMD_NODENAME}, submitted from \${SLURM_SUBMIT_HOST}" && sleep 60 && echo "Job
complete"
EOF
```

- 將任務指令碼提交至 Slurm 排程器：

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

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

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

Example

```
squeue --job 1
```

`squeue` 命令會傳回類似下列的輸出：

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

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

```
ls -alth /shared
```

命令輸出類似於以下內容：

```
-rw-rw-r- 1 ec2-user ec2-user 107 Mar 19 18:33 single.1.out
-rw-rw-r- 1 ec2-user ec2-user 0 Mar 19 18:32 single.1.err
```

```
-rw-rw-r- 1 ec2-user ec2-user 381 Mar 19 18:29 job.sh
```

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

- 檢查 `single.1.out` 檔案的內容。

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

輸出類似以下內容：

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

在 AWS PCS 中使用 Slurm 執行多節點 MPI 任務

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

在登入節點的 shell 提示下執行下列命令。

- 成為預設使用者。變更為其主目錄。

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

- 使用 C 程式設計語言建立原始程式碼。

```
cat > hello.c << EOF
// * mpi-hello-world - https://www.mpitutorial.com
// Released under MIT License
//
// Copyright (c) 2014 MPI Tutorial.
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to
// deal in the Software without restriction, including without limitation the
// rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
// sell copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
```

```
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.

#include <mpi.h>
#include <stdio.h>
#include <stddef.h>

int main(int argc, char** argv) {
    // Initialize the MPI environment. The two arguments to MPI Init are not
    // currently used by MPI implementations, but are there in case future
    // implementations might need the arguments.
    MPI_Init(NULL, NULL);

    // Get the number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    // Get the name of the processor
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    // Print off a hello world message
    printf("Hello world from processor %s, rank %d out of %d processors\n",
           processor_name, world_rank, world_size);

    // Finalize the MPI environment. No more MPI calls can be made after this
    MPI_Finalize();
}
EOF
```

- 載入 OpenMPI 模組。

```
module load openmpi
```

- 編譯 C 程式。

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

- 撰寫 Slurm 任務提交指令碼。

```
cat > hello.sh << EOF
#!/bin/bash
#SBATCH -J multi
#SBATCH -o multi.out
#SBATCH -e multi.err
#SBATCH --exclusive
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1

srun $HOME/hello
EOF
```

- 變更為共用目錄。

```
cd /shared
```

- 提交任務指令碼。

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

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

```
cat multi.out
```

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

```
Hello world from processor ip-10-3-133-204, rank 0 out of 4 processors
Hello world from processor ip-10-3-128-219, rank 2 out of 4 processors
Hello world from processor ip-10-3-141-26, rank 3 out of 4 processors
Hello world from processor ip-10-3-143-52, rank 1 out of 4 processor
```

刪除 AWS PCS AWS 的資源

完成您為本教學課程建立的叢集和節點群組後，您應該刪除您建立的資源。

Important

您可以針對 中執行的所有資源收取帳單費用 AWS 帳戶

刪除您為此教學課程建立的 AWS PCS 資源

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

Important

等待佇列刪除後再繼續。

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

Important

請等到兩個運算節點群組都已刪除，再繼續。


- 在入門的叢集詳細資訊頁面中，選擇刪除。

Important


等待叢集刪除後，再繼續後續步驟。

刪除您為此教學課程建立的其他 AWS 資源


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

 Important
等待堆疊刪除再繼續。

- 開啟 [Amazon EFS 主控台](#)。
 - 選擇檔案系統。
 - 選取名為 getstarted-efs 的檔案系統。
 - 選擇 刪除。

 Important
等待檔案系統刪除，然後再繼續。

- 開啟 [Amazon FSx 主控台](#)。
 - 選擇檔案系統。
 - 選取名為 getstarted-fsx 的檔案系統。
 - 選擇 刪除。

 Important
等待檔案系統刪除，然後再繼續。

- 開啟 [CloudFormation 主控台](#)。
 - 選取名為 getstarted-sg 的堆疊。

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

CloudFormation 和 AWS PCS 入門

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

主題

- [使用 CloudFormation 建立範例 AWS PCS 叢集](#)
- [連線至使用建立的 AWS PCS 叢集 CloudFormation](#)
- [在中清除 AWS PCS 叢集 CloudFormation](#)
- [AWS PCS CloudFormation 範本的一部分](#)
- [CloudFormation 建立範例 AWS PCS 叢集的範本](#)

使用 CloudFormation 建立範例 AWS PCS 叢集

下列程序使用中的 CloudFormation 範本 AWS 管理主控台 來建立範例 AWS PCS 叢集。如需詳細資訊 CloudFormation，請參閱[什麼是 CloudFormation？](#) AWS CloudFormation 《使用者指南》中的。如需中 AWS PCS 資源類型的詳細資訊 CloudFormation，請參閱AWS CloudFormation 《使用者指南》中的[AWS PCS 資源類型參考](#)。

建立範例叢集

1. 選擇要在中 AWS 區域 建立叢集的（連結會開啟具有範本的 CloudFormation 主控台）：
 - [US East \(N. Virginia\) \(美國東部 \(維吉尼亞北部\)\) \(us-east-1\)](#)
 - [US East \(Ohio\) \(美國東部 \(俄亥俄\)\) \(us-east-2\)](#)
 - [US West \(Oregon\) \(美國西部 \(奧勒岡\)\) \(us-west-2\)](#)
 - [Asia Pacific \(Mumbai\) \(亞太區域 \(孟買\)\) \(ap-south-1\)](#)
 - [Asia Pacific \(Singapore\) \(亞太區域 \(新加坡\)\) \(ap-southeast-1\)](#)
 - [Asia Pacific \(Sydney\) \(亞太區域 \(雪梨\)\) \(ap-southeast-2\)](#)

- [Asia Pacific \(Tokyo\) \(亞太區域 \(東京\)\) \(ap-northeast-1\)](#)
 - [歐洲 \(法蘭克福\) \(eu-central-1\)](#)
 - [歐洲 \(愛爾蘭\) \(eu-west-1\)](#)
 - [歐洲 \(倫敦\) \(eu-west-2\)](#)
 - [歐洲 \(巴黎\) \(eu-west-3\)](#)
 - [歐洲 \(米蘭\) \(eu-south-1\)](#)
 - [歐洲 \(斯德哥爾摩\) \(eu-north-1\)](#)
 - [AWS GovCloud \(美國東部\) \(us-gov-east-1\)](#)
 - [AWS GovCloud \(美國西部\) \(us-gov-west-1\)](#)
2. 在提供堆疊名稱下，輸入描述性名稱。這是 CloudFormation 堆疊的名稱。範本使用此值做為 AWS PCS 叢集的名稱。
 3. 在參數下：
 - a. 在 SlurmVersion 下，選擇您要叢集使用的 Slurm 版本。
 - b. 在 NodeArchitecture 下，選擇 x86 以部署使用 x86_64 相容執行個體的叢集，或選擇 Graviton 以使用 Arm64 執行個體。
 - c. 針對 KeyName，選擇 SSH 金鑰對以存取叢集登入節點。請確定您擁有所選金鑰對的 PEM 檔案。
 - d. 針對 ClientIpCidr，以 CIDR 格式輸入 IP 範圍，以控制對登入節點的存取。
-  **Warning**
的預設值 0.0.0.0/0 允許從所有 IP 地址存取。
- e. 將 HpcRecipesS3Bucket 和 HpcRecipesBranch 的值保留為預設值。
4. 在功能和轉換下：
 - a. 選取核取方塊以確認 CloudFormation 將建立 IAM 資源。
 - b. 選取核取方塊以確認 CloudFormation 將使用自訂名稱建立 IAM 資源。
 - c. 選取核取方塊以確認 CAPABILITY_AUTO_EXPAND 新堆疊。如需詳細資訊，請參閱 AWS CloudFormation API 參考中的 [CreateStack](#)。
5. 選擇建立堆疊。
6. 監控堆疊的狀態。您可以在堆疊的狀態為 之後連線到叢集 CREATE_COMPLETE。

連線至使用建立的 AWS PCS 叢集 CloudFormation

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

連線至您的叢集

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

堆疊提供下列連結：

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

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

連線至登入執行個體後，您可以遵循的教學[探索 AWS PCS 中的叢集環境](#)課程。

在 中清除 AWS PCS 叢集 CloudFormation

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

Important

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

AWS PCS CloudFormation 範本的一部分

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

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

Important

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

內容

- [標頭](#)
- [中繼資料](#)
- [Parameters](#)
- [映射項目](#)
- [Resources](#)
- [輸出](#)

標頭

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

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

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

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

中繼資料

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

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

Parameters

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

```
Parameters:

  NodeArchitecture:
    Type: String
    Default: x86
```

AllowedValues:

- x86
- Graviton

Description: Processor architecture for the login and compute node instances

SlurmVersion:

Type: String

Default: 25.05

Description: Version of Slurm to use

AllowedValues:

- 24.11
- 25.05

ManagedAccounting:

Type: String

Default: 'disabled'

AllowedValues:

- 'enabled'
- 'disabled'

Description: Monitor cluster usage, manage access control, and enforce resource limits with Slurm accounting. Requires Slurm 24.11 or newer.

AccountingPolicyEnforcement:

Description: Specify which Slurm accounting policies to enforce

Type: String

Default: none

AllowedValues:

- none
- 'associations,limits,safe'

KeyName:

Description: SSH keypair to log in to the head node

Type: AWS::EC2::KeyPair::KeyName

AllowedPattern: ".+" # Required

ClientIpCidr:

Description: IP(s) allowed to access the login node over SSH. We recommend that you restrict it with your own IP/subnet (x.x.x.x/32 for your own ip or x.x.x.x/24 for range. Replace x.x.x.x with your own PUBLIC IP. You can get your public IP using tools such as <https://ifconfig.co/>)

Default: 127.0.0.1/32

Type: String

AllowedPattern: (\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3})/(\d{1,2})

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

HpcRecipesS3Bucket:

```
Type: String
Default: aws-hpc-recipes
Description: HPC Recipes for AWS S3 bucket
AllowedValues:
  - aws-hpc-recipes
  - aws-hpc-recipes-dev
```

HpcRecipesBranch:

```
Type: String
Default: main
Description: HPC Recipes for AWS release branch
AllowedPattern: '^(?!.*\/\.$)(?!.*\/\.)[a-zA-Z0-9-_\.\.]+$'
```

映射項目

Mappings 區段定義鍵/值對，根據特定條件或相依性指定值。

Mappings:

Architecture:

```
AmiArchParameter:
  Graviton: arm64
  x86: x86_64
```

LoginNodeInstances:

```
Graviton: c7g.xlarge
x86: c6i.xlarge
```

ComputeNodeInstances:

```
Graviton: c7g.xlarge
x86: c6i.xlarge
```

Resources

Resources 區段會宣告要佈建和設定作為堆疊一部分 AWS 的資源。

Resources:

```
[...]
```

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

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

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

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

```
[...]
```

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

```

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

```

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

```

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

PCSLaunchTemplate:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      VpcDefaultSecurityGroupId: !GetAtt [ Networking, Outputs.SecurityGroup ]
      ClusterSecurityGroupId: !GetAtt [ PCSSecurityGroup,
Outputs.ClusterSecurityGroupId ]
      SshSecurityGroupId: !GetAtt [ PCSSecurityGroup,
Outputs.InboundSshSecurityGroupId ]
      EfsFileSystemSecurityGroupId: !GetAtt [ EfsStorage, Outputs.SecurityGroupId ]
      FSxLustreFileSystemSecurityGroupId: !GetAtt [ FSxLStorage,
Outputs.FSxLustreSecurityGroupId ]
      SshKeyName: !Ref KeyName
      EfsFileSystemId: !GetAtt [ EfsStorage, Outputs.EFSFileSystemId ]
      FSxLustreFileSystemId: !GetAtt [ FSxLStorage, Outputs.FSxLustreFileSystemId ]
      FSxLustreFileSystemMountName: !GetAtt [ FSxLStorage,
Outputs.FSxLustreMountName ]

```

```
TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/cfn-pcs-1t-efs-fsxl.yaml'

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

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

任務排程是透過 處理 PCSQueueCompute。

```

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

```

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

```

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

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

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

```

輸出

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

Outputs:**ClusterId:**

Description: The Id of the PCS cluster

Value: !GetAtt [PCSCluster, Id]

PcsConsoleUrl:

Description: URL to access the cluster in the PCS console

Value: !Sub

- https://\${ConsoleDomain}/pcs/home?region=\${AWS::Region}#/clusters/\${ClusterId}

- { ConsoleDomain: !If [GovCloud, 'console.amazonaws-us-gov.com', !If [China, 'console.amazonaws.cn', !Sub '\${AWS::Region}.console.aws.amazon.com']],

ClusterId: !GetAtt [PCSCluster, Id]

}

Export:

Name: !Sub \${AWS::StackName}-PcsConsoleUrl

Ec2ConsoleUrl:

Description: URL to access instance(s) in the login node group via Session Manager

Value: !Sub

- https://\${ConsoleDomain}/ec2/home?region=

\${AWS::Region}#Instances:instanceState=running;tag:aws:pcs:compute-node-group-id=\${NodeGroupLoginId}

- { ConsoleDomain: !If [GovCloud, 'console.amazonaws-us-gov.com', !If [China, 'console.amazonaws.cn', !Sub '\${AWS::Region}.console.aws.amazon.com']],




NodeGroupLoginId: !GetAtt [PCSNodeGroupLogin, Id]

}

Export:

Name: !Sub \${AWS::StackName}-Ec2ConsoleUrl

CloudFormation 建立範例 AWS PCS 叢集的 範本

AWS 區域 名稱	AWS 區域	檢視來源	啟動堆疊
美國東部 (維吉尼亞北部)	us-east-1	下載 YAML	
美國東部 (俄亥俄)	us-east-2	下載 YAML	
美國西部 (奧勒岡)	us-west-2	下載 YAML	

AWS 區域 名稱	AWS 區域	檢視來源	啟動堆疊
亞太區域 (孟買)	ap-south-1	下載 YAML	
亞太區域 (新加坡)	ap-southeast-1	下載 YAML	
亞太地區 (悉尼)	ap-southeast-2	下載 YAML	
亞太區域 (東京)	ap-northeast-1	下載 YAML	
歐洲 (法蘭克福)	eu-central-1	下載 YAML	
歐洲 (愛爾蘭)	eu-west-1	下載 YAML	
歐洲 (倫敦)	eu-west-2	下載 YAML	
Europe (Paris)	eu-west-3	下載 YAML	
歐洲 (米蘭)	eu-south-1	下載 YAML	
Europe (Stockholm)	eu-north-1	下載 YAML	
AWS GovCloud (美國東部)	us-gov-east-1	下載 YAML	
AWS GovCloud (美國西部)	us-gov-west-1	下載 YAML	

AWS PCS 叢集

AWS PCS 叢集包含下列元件：

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

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

主題

- [在 AWS PCS 中建立叢集](#)
- [在 AWS PCS 中更新叢集](#)
- [在 AWS PCS 中刪除叢集](#)
- [AWS PCS 中的叢集大小](#)
- [在 AWS PCS 中使用叢集秘密](#)

在 AWS PCS 中建立叢集

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

Note

建立叢集之後，您可以修改許多組態設定，而無需重建基礎設施。如需詳細資訊，請參閱在 [AWS PCS 中更新叢集](#)。

Note

您可以設定自訂 Slurm 設定，以實作進階排程政策和資源管理。如需詳細資訊，請參閱在 [AWS PCS 中設定自訂 Slurm 設定](#)。

先決條件

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

建立 AWS PCS 叢集

您可以使用 AWS 管理主控台 或 AWS CLI 來建立叢集。

AWS 管理主控台

建立叢集

1. 在 <https://console.aws.amazon.com/pcs/home#/clusters> 開啟 AWS PCS 主控台，然後選擇建立叢集。
2. 在叢集設定區段中，輸入下列欄位：
 - 叢集名稱 – 叢集的名稱。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 40 個字元。名稱在 中必須是唯一的 AWS 帳戶，AWS 區域 且您要在其中建立叢集。
 - 排程器 – 選擇排程器和版本。如需詳細資訊，請參閱 [AWS PCS 中的 Slurm 版本](#)。
 - 控制器大小 – 選擇控制器的大小。這決定 AWS PCS 叢集可以管理多少並行任務和運算節點。您只能在建立叢集時設定控制器大小。如需調整大小的詳細資訊，請參閱 [AWS PCS 中的叢集大小](#)。
3. 在聯網區段中，選取下列欄位的值：
 - 網路類型 – 選擇叢集的 IP 地址類型。您的叢集可以使用 IPv4 或 IPv6，但不能同時使用兩者。VPC 和子網路必須使用相同的網路地址類型。您用於每個子網路的 IP 地址區塊必

須至少有 1 個可用地址。AWS 會保留每個子網路中的部分地址。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[子網路 CIDR 區塊](#)。

- VPC – 選擇符合 AWS PCS 要求的現有 VPC。如需詳細資訊，請參閱[AWS PCS VPC 和子網路需求和考量事項](#)。建立叢集之後，您無法變更其 VPC。如果未列出 VPCs，您必須先建立一個 VPC。
 - 子網路 – 列出所選 VPC 中的所有可用子網路。選擇符合 AWS PCS 子網路需求的子網路。如需詳細資訊，請參閱[AWS PCS VPC 和子網路需求和考量事項](#)。我們建議您選取私有子網路，以避免您的排程器端點暴露至公有網際網路。
 - 安全群組 – 指定您希望 AWS PCS 與其為叢集建立的網路介面建立關聯的安全群組（多個）。您必須選取至少一個安全群組，以允許叢集及其運算節點之間的通訊。您可以選取快速建立安全群組，讓 AWS PCS 在您選取的 VPC 中建立具有必要組態的安全群組，或選取現有的安全群組。如需詳細資訊，請參閱[安全群組需求和考量事項](#)。
4. （選用）在 Slurm 會計組態區段中，您可以啟用 Slurm 會計並設定會計參數。如需詳細資訊，請參閱[AWS PCS 中的 Slurm 會計](#)。
 5. （選用）在 Slurm 組態區段中，您可以新增參數名稱和值對來設定其他 Slurm 設定。如需支援參數的完整清單，請參閱 [AWS PCS 叢集的自訂 Slurm 設定](#)。
 6. （選用）在標籤下，將任何標籤新增至 AWS PCS 叢集。
 7. 選擇 Create Cluster (建立叢集)。當 AWS PCS 建立叢集 Creating 時，狀態欄位會顯示。此程序需要幾分鐘的時間。

Important

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

AWS CLI

建立叢集

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

- 將 **25.05** 取代為任何支援的 Slurm 版本。

Note

AWS PCS 目前支援 Slurm 25.05 和 24.11。

- 將 **SMALL** 取代為任何支援的叢集大小。這決定 AWS PCS 叢集可以管理多少並行任務和運算節點。它只能在建立叢集時設定。如需調整大小的詳細資訊，請參閱[AWS PCS 中的叢集大小](#)。
- 將 `subnetIds` 的值取代為您自己的值。我們建議您選取私有子網路，以避免您的排程器端點暴露至公有網際網路。
- 指定 `securityGroupIds` 您希望 AWS PCS 與其為叢集建立的網路介面建立關聯的。安全群組必須與叢集位於相同的 VPC 中。您必須選取至少一個安全群組，以允許叢集及其運算節點之間的通訊。如需詳細資訊，請參閱[安全群組需求和考量事項](#)。

```
aws pcs create-cluster --region region \  
  --cluster-name my-cluster \  
  --scheduler type=SLURM,version=25.05 \  
  --size SMALL \  
  --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

- 若要使用 IPv6，請將 `networkType=IPV6` 新增至 `--networking` 組態。

```
--networking networkType=IPV6,subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

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

Note

Slurm 24.11 或更新版本支援會計。

```
aws pcs create-cluster --region region \  
  --cluster-name my-cluster \  
  --scheduler type=SLURM,version=25.05 \  
  --size SMALL \  
  --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1 \  
  --slurm-configuration  
  scaleDownIdleTimeInSeconds=3600,accounting='{mode=STANDARD}',slurmCustomSettings='{p
```

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

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

Important

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

您叢集的建議後續步驟

- 新增運算節點群組。
- 新增佇列。
- 啟用記錄。

在 AWS PCS 中更新叢集

AWS PCS 可讓您在建立叢集組態後，透過 `UpdateCluster` API 或主控台更新叢集組態。您可以在不重建基礎設施的情況下修改叢集設定，從而減少營運開銷並將中斷降至最低。

叢集更新的優點

更新 AWS PCS 叢集可讓您根據新需求調整 HPC 基礎設施，而不會中斷服務。組態變更需要幾分鐘的時間，而不是重建叢集所需的小時或更長的時間。對於需要最短停機時間的生產環境，以及需要隨著工作負載模式變更調整叢集設定的團隊而言，此功能很重要。

支援的組態變更

您可以修改三個主要類別的設定：

- 會計組態 - 啟用或停用受管會計並設定保留設定。
- 縮減規模行為 - 調整 `scaleDownIdleTime` 參數，以控制動態執行個體在 AWS PCS 自動終止它們之前保持閒置狀態的時間。
- Slurm 自訂設定 - 修改叢集層級套用的任何支援 Slurm 設定，包括 Prolog、Epilog 和 `SelectTypeParameters`。

限制

您無法在建立叢集後修改特定組態。其中包含：

- 安全群組組態
- VPC 子網路選擇
- 叢集大小
- Slurm 版本
- 叢集名稱

這些設定是叢集架構的基礎，需要建立新的叢集來修改它們。

叢集更新的先決條件

更新叢集之前，請確定符合下列條件：

- 叢集必須處於 ACTIVE、UPDATE_FAILED 或 SUSPENDED 狀態
- 所有相關資源（佇列、運算節點群組）必須處於 ACTIVE 狀態
- 您必須擁有 UpdateCluster 操作的適當 IAM 許可
- 無法進行其他更新操作

更新程序和任務影響

在更新操作期間，即使叢集控制器短暫無法連線，運算節點仍會繼續執行現有任務。不過，系統在此期間無法接受新的任務提交或做出排程決策。

您可以透過主控台和 API 界面來監控叢集更新。叢集將在更新期間轉換下列狀態：

- UPDATING - 更新進行中
- ACTIVE - 更新成功完成
- UPDATE_FAILED - 更新發生錯誤

在更新期間計費

AWS PCS 叢集的標準每小時費用會在更新操作期間繼續。當您更新叢集以停用會計時，一旦叢集進入 UPDATING 狀態，會計功能的帳單就會停止。啟用會計時，在叢集成功完成更新並返回 ACTIVE 狀態之前，不會開始計費。

主題

- [更新 AWS PCS 叢集](#)
- [有關在 AWS PCS 中更新叢集的常見問答集](#)
- [疑難排解 AWS PCS 叢集更新](#)

更新 AWS PCS 叢集

使用這些步驟來修改叢集上的排程器設定、會計組態和 Slurm 自訂設定。如需詳細資訊，請參閱[AWS PCS 叢集的自訂 Slurm 設定](#)。

先決條件


- 叢集必須處於 ACTIVE、UPDATE_FAILED 或 SUSPENDED 狀態
- 所有相關資源（佇列、運算節點群組）必須處於 ACTIVE 狀態
- 無法進行其他更新操作

程序

AWS 管理主控台

1. 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選取要更新的叢集。

4. 選擇編輯。
5. 在編輯叢集頁面上，修改所需的設定：
 - 在排程器組態下，更新縮減閒置時間，以控制動態執行個體在自動終止前保持閒置的時間長度。
 - 視需要修改 Prolog、Epilog 和 Select 類型參數設定。
 - 啟用、停用或設定受管會計的保留時間。
 - 在其他排程器設定下，新增、編輯或移除 Slurm 自訂設定。如需支援參數的詳細資訊，請參閱 [AWS PCS 叢集的自訂 Slurm 設定](#)。

 Note

無法編輯的欄位會顯示為唯讀，並顯示其目前的值。

6. 選擇更新以提交變更。
7. 監控叢集狀態，在程序期間顯示為「更新」。當更新成功完成時，狀態會變更。

AWS CLI

1. 開啟終端機或命令提示字元。
2. 使用以下命令驗證叢集狀態：

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

3. 使用下列其中一個範例提交更新請求：

- 若要啟用受管會計：

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration 'accounting={mode=STANDARD}'
```

- 若要更新 Slurm Prolog 設定：

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/  
prolog.sh"}]'
```

- 若要更新縮減規模閒置時間：

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration 'scaleDownIdleTimeInSeconds=300'
```

4. 檢查叢集狀態以監控更新進度：

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

成功更新請求後，命令會傳回包含所有變更的叢集物件。叢集狀態會在完成UPDATINGACTIVE時從變更為。

有關在 AWS PCS 中更新叢集的常見問答集

取得有關在 AWS PCS 中更新叢集組態的常見問題解答。

我可以修改哪些設定？

您可以修改會計組態（啟用/停用受管會計）、縮減規模行為（scaleDownIdleTime 參數），以及叢集層級套用的任何支援的 Slurm 自訂設定。您無法修改安全群組、VPC 子網路、叢集大小、Slurm 版本或叢集名稱。

我可以將多個更新排入佇列嗎？

否。在提交另一個更新之前，您必須等待叢集返回 ACTIVE 狀態。所有相關資源（佇列、運算節點群組）也必須處於 ACTIVE 狀態。

我可以取消叢集更新操作嗎？

否，您無法取消正在進行的叢集更新操作。

我可以在叢集更新時提交任務嗎？

建議您避免在叢集更新期間提交任務。在更新過程中，Slurm 控制器可能無法使用。

在叢集更新期間，我的任務是否會繼續執行？

是，即使叢集控制器在更新程序期間變得短暫無法連線，執行中的任務仍會繼續在運算節點上執行。不過，在控制器再次可用之前，任務狀態可能不會更新。

在更新期間，帳單會受到什麼影響？

在更新操作期間，會繼續收取標準每小時費用。停用會計時，當叢集進入 UPDATING 狀態時，計費會停止。啟用會計時，帳單會在叢集成功返回 ACTIVE 狀態時開始。

疑難排解 AWS PCS 叢集更新

本主題可協助您識別並解決更新叢集組態時可能發生的常見問題。

更新失敗，會計組態錯誤

常見原因

叢集進入 UPDATE_FAILED 狀態，錯誤訊息指出會計組態問題。這通常發生在會計組態與目前的 Slurm 版本不相容或包含無效的設定時。

Resolution

檢閱您的會計設定是否與叢集的 Slurm 版本相容，並使用有效的組態參數提交更正後的更新請求。

更新失敗，並出現自訂設定錯誤

常見原因

叢集進入 UPDATE_FAILED 狀態，錯誤訊息指出 Slurm 自訂設定問題。當您提供無效的 Slurm 參數值或不支援的參數組合時，就會發生這種情況。

Resolution

根據支援的參數驗證 Slurm 自訂設定，並提交具有有效參數值和組合的已更正更新請求。

無法提交更新請求

常見原因

更新按鈕會在主控台中停用，否則 API 會傳回 400 層級的錯誤。當叢集未處於適當狀態、相關聯的資源未處於作用中狀態，或您的組態中有驗證失敗時，就會發生這種情況。

Resolution

等待叢集和所有相關資源達到 ACTIVE 狀態，然後檢閱您的組態是否有驗證錯誤，再重新提交更新請求。

驗證錯誤

常見原因

命令會立即傳回 400 層級 HTTP 錯誤和描述性訊息。這是因為無效的叢集狀態、資源狀態或組態參數。

Resolution

解決回應中提到的特定驗證錯誤，然後重試更新操作。

在 AWS PCS 中刪除叢集

本主題提供如何刪除 AWS PCS 叢集的概觀。

刪除 AWS PCS 叢集時的考量事項

- 必須先刪除與叢集相關聯的所有佇列，才能刪除叢集。如需詳細資訊，請參閱[在 AWS PCS 中刪除佇列](#)。
- 必須先刪除與叢集相關聯的所有運算節點群組，才能刪除叢集。如需詳細資訊，請參閱[在 AWS PCS 中刪除運算節點群組](#)。

刪除叢集

您可以使用 AWS 管理主控台 或 AWS CLI 來刪除叢集。

AWS 管理主控台

刪除叢集

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

AWS CLI

刪除叢集

1. 使用下列命令來刪除具有這些替換項目的叢集：
 - AWS 區域 將 *region-code* 取代為叢集所在的。
 - 以叢集的名稱或 ID 取代 *my-cluster*。

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

2. 刪除叢集可能需要幾分鐘的時間。您可以使用下列命令來檢查叢集的狀態。

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

AWS PCS 中的叢集大小

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

當您建立叢集時，您可以根據兩個因素為其選取大小：

- 將管理的運算節點數量
- 控制器在任何指定時間追蹤的任務數量

Note

任務計數包括執行中、待定和最近完成的任務。在清除之前，控制器會追蹤已完成的任務一小段時間。在高任務輸送量期間，這可能會導致追蹤的任務總數超過您觀察到的作用中任務數量。

Important

您無法在建立叢集之後變更叢集大小。如果您需要變更大小，則必須建立新的叢集。

Slurm 叢集大小	受管執行個體數量	控制器追蹤的任務數量
小型	最多 32 個	最多 256 個
中	最多 512 個	高達 8192
大型	截至 2048 年	高達 16384

範例

- 如果您的叢集最多會有 24 個受管執行個體並執行最多 100 個任務，請選擇小型。
- 如果您的叢集最多會有 24 個受管執行個體並執行最多 1000 個任務，請選擇中。
- 如果您的叢集最多會有 1000 個受管執行個體並執行最多 100 個任務，請選擇大型。
- 如果您的叢集最多會有 1000 個受管執行個體，並執行最多 10,000 個任務，請選擇大型。

在 AWS PCS 中使用叢集秘密

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

AWS PCS 會將叢集秘密儲存為 pcs! 中字首為的 [受管秘密](#) AWS Secrets Manager。秘密的成本包含在使用 AWS PCS 的費用中。您可以透過 [輪換叢集秘密 AWS Secrets Manager](#)，以維持安全合規並修復潛在的安全危害。

主題

- [使用 AWS Secrets Manager 尋找叢集秘密](#)
- [使用 AWS PCS 尋找叢集秘密](#)
- [取得 Slurm 叢集秘密](#)
- [在 AWS PCS 中輪換叢集秘密](#)

使用 AWS Secrets Manager 尋找叢集秘密

AWS 管理主控台

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

Note

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

AWS CLI

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

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

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

使用 AWS PCS 尋找叢集秘密

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

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

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

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

```
{
  "cluster": {
    "name": "get-started",
    "id": "pcs_123456abcd",
    "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
    "status": "ACTIVE",
    "createdAt": "2024-12-17T21:03:52+00:00",
    "modifiedAt": "2024-12-17T21:03:52+00:00",
    "scheduler": {
      "type": "SLURM",
      "version": "25.05"
    },
    "size": "SMALL",
    "slurmConfiguration": {
      "authKey": {
        "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!slurm-secret-pcs_123456abcd-a12ABC",
        "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
      }
    },
    "networking": {
      "subnetIds": [
        "subnet-0123456789abcdef0"
      ],
      "securityGroupIds": [
        "sg-0123456789abcdef0"
      ]
    },
    "endpoints": [
      {
        "type": "SLURMCTLD",
        "privateIpAddress": "10.3.149.220",
        "port": "6817"
      }
    ]
  }
}
```

取得 Slurm 叢集秘密

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

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

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

如需如何使用 Slurm 叢集秘密的資訊，請參閱 [使用獨立執行個體做為 AWS PCS 登入節點](#)。

許可

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

下列範例 IAM 政策允許存取範例叢集秘密。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowSecretValueRetrievalAndVersionListing",  
      "Effect": "Allow",  
      "Action": [  
        "secretsmanager:GetSecretValue",  
        "secretsmanager:ListSecretVersionIds"  
      ],  
      "Resource": "arn:aws:secretsmanager:us-east-1:012345678901:secret:pcs!  
slurm-secret-s3431v9rx2-FN7tJF"  
    }  
  ]  
}
```

在 AWS PCS 中輪換叢集秘密

使用 AWS Secrets Manager 受管輪換在 AWS PCS 中輪換叢集秘密。定期秘密輪換是維護 HPC 環境中強式安全狀態的安全最佳實務。此功能可讓您符合產業合規標準，包括 HIPAA 和 FedRAMP，這需要定期輪換登入資料。

叢集秘密提供雙重用途：驗證加入叢集的運算節點，以及做為 Slurm REST API 身分驗證的 JWT 金鑰。輪換時，兩個層面都會同時受到影響。

叢集秘密輪換的運作方式

手動準備以在秘密輪換期間維持叢集穩定性：

1. 準備 – 將所有運算節點群組擴展到 0 容量，並確保沒有任務正在執行
2. 輪換 – 透過 Secrets Manager 主控台或 API 啟動輪換
3. 監控 – 透過 CloudTrail 事件追蹤進度
4. 復原 – 將運算節點群組擴展回所需的容量

在輪換期間，您的叢集會保持 ACTIVE 狀態，且計費會繼續正常。程序通常需要幾分鐘的時間。

要求與限制

在輪換叢集秘密之前，請先完成下列要求：

- 叢集必須處於 ACTIVE 或 UPDATE_FAILED 狀態
- IAM 角色必須具有 `secretsmanager:RotateSecret` 許可
- 所有運算節點群組都必須擴展至 0 容量
- 在輪換之前停止所有任務

限制：

- 每次輪換所需的手動準備
- 現有的 JWT 字符無效，需要重新發行
- BYO 登入節點在輪換後需要手動秘密更新

主題

- [在 AWS PCS 中輪換叢集秘密](#)
- [AWS PCS 中叢集秘密輪換的常見問題](#)
- [對 AWS PCS 中的叢集秘密輪換進行故障診斷](#)

在 AWS PCS 中輪換叢集秘密

輪換您的叢集秘密以符合安全要求，並解決潛在的入侵。此程序需要讓您的叢集進入維護模式。

先決條件

- 具有 `secretsmanager:RotateSecret` 許可的 IAM 角色
- 叢集處於 ACTIVE 或 UPDATE_FAILED 狀態

程序

1. 通知叢集使用者即將到來的維護時段。
2. 透過將所有運算節點群組擴展到 0 容量，將叢集置於維護模式。
 - a. 使用 `UpdateComputeNodeGroup` API，將所有運算節點群組的 `minInstanceCount` 和 `maxInstanceCount` 設定為 0。
 - b. 等到所有節點停止。
 - c. 選用：在您終止容量以進行正常任務處理之前，使用 `Slurm` 命令耗盡排程器佇列。
3. 透過 Secrets Manager 啟動輪換。
 - 主控台方法：
 - 導覽至 Secrets Manager，選取叢集秘密，然後選擇輪換秘密。
 - API 方法：
 - 使用 Secrets Manager `rotate-secret` API。
4. 監控輪換進度。
 - a. 透過 CloudTrail 事件追蹤進度。
 - b. `lastRotatedDate` 檢查 Secrets Manager 主控台或 `secretsmanager:describeSecret` API。
 - c. 等待 `RotationSucceeded` 或 `RotationFailed` CloudTrail 事件。
5. 輪換成功後，請還原叢集容量。
 - a. 使用 `UpdateComputeNodeGroup` API 將節點群組重設為所需的最小/最大容量。
 - b. 對於 AWS PCS 受管登入節點：不需要其他動作。
 - c. 對於 BYO 登入節點：

- i. 連線至登入節點。
- ii. `/etc/slurm/slurm.key` 使用 Secrets Manager 的新秘密進行更新。
- iii. 重新啟動 Slurm Auth 和 Cred Kiosk 協助程式（已封裝）。

AWS PCS 中叢集秘密輪換的常見問題

尋找 AWS PCS 中叢集秘密輪換的常見問題解答。

什麼是叢集秘密？

叢集秘密是一種安全登入資料，可啟用 Slurm 控制器和 AWS PCS 運算節點之間的安全通訊。它也可以做為 Slurm REST API 身分驗證的 JSON Web Token (JWT) 金鑰。

叢集秘密和 JWT 金鑰之間的差異是什麼？

在 AWS PCS 中，叢集秘密和 JWT 金鑰是提供不同用途的相同資源。叢集秘密會驗證 Slurm 內部通訊，而 JWT 金鑰會簽署字符以進行 REST API 身分驗證。輪換時，兩個層面都會同時受到影響。

輪換需要多長時間？

輪換程序通常需要幾分鐘的時間。您的叢集會保持 ACTIVE 狀態，並在輪換期間繼續正常計費。

我可以排程自動輪換嗎？

您可以在 Secrets Manager 中啟用排程輪換。不過，初始版本需要在每次輪換之前手動準備（將節點群組擴展至 0）。

我現有的 JWT 權杖在輪換後仍可運作嗎？

否，現有的 JWT 權杖在輪換後會變成無效。為 REST API 用戶端發行新權杖。

哪裡可以找到我的叢集秘密？

您可以在 Secrets Manager 主控台或透過 AWS PCS 主控台找到叢集秘密。如需詳細說明，請參閱 [使用 AWS Secrets Manager 尋找叢集秘密](#) 和 [使用 AWS PCS 尋找叢集秘密](#)。

為什麼輪換需要將節點群組擴展至 0？

輪換不需要執行中的執行個體，以確保秘密更新程序期間的叢集穩定性。這可防止舊秘密和新秘密之間的身分驗證衝突。

此功能支援哪些合規要求？

此功能可讓 AWS PCS 符合產業合規標準，包括 HIPAA 和 FedRAMP，其要求定期輪換登入資料，作為其安全控制的一部分。

對 AWS PCS 中的叢集秘密輪換進行故障診斷

如果環境未正確準備，叢集秘密輪換會失敗。最常見的原因是叢集中的作用中執行個體。若要防止失敗：

1. 將所有節點群組設定為 0 容量。
2. 等待節點停止。
3. 確認您的叢集未處於這些狀態：CREATE_FAILED、DELETE_FAILED、SUSPENDING、RESUMING 或 SUSPENDED。

如果輪換失敗：

- 隨即出現 RotationFailed CloudTrail 事件
- 叢集秘密保持不變
- 如需詳細資訊，請檢查 CloudTrail 中的 RotationFailed 事件
- 完成所有準備步驟以成功輪換

AWS PCS 運算節點群組

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

主題

- [在 AWS PCS 中建立運算節點群組](#)
- [更新 AWS PCS 運算節點群組](#)
- [在 AWS PCS 中刪除運算節點群組](#)
- [在 AWS PCS 中取得運算節點群組詳細資訊](#)
- [尋找 AWS PCS 中的運算節點群組執行個體](#)

在 AWS PCS 中建立運算節點群組

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

Note

您可以在運算節點群組上設定自訂 Slurm 設定，以控制資源使用率和節點層級行為。如需詳細資訊，請參閱 [在 AWS PCS 中設定自訂 Slurm 設定](#)。

Important

AWS PCS 目前需要支援 IPv4 的核心進行本機節點通訊，即使您在 IPv6-only 的網路中使用 AWS PCS 也是如此。如需詳細資訊，請參閱 [AWS PCS 的自訂 Amazon Machine Image AMIs](#)。

先決條件

- 有足夠的服務配額，可在您的 中啟動所需數量的 EC2 執行個體 AWS 區域。您可以使用 [AWS 管理主控台](#) 來檢查和請求提高您的服務配額。
- 符合 AWS PCS 聯網需求的現有 VPC 和子網路 (s)。建議您在部署叢集以供生產使用之前，先徹底了解這些需求。如需詳細資訊，請參閱 [AWS PCS VPC 和子網路需求和考量事項](#)。您也可以使用 CloudFormation 範本來建立 VPC 和子網路。為 CloudFormation 範本 AWS 提供 HPC 配方。如需詳細資訊，請參閱 GitHub 上的 [aws-hpc-recipes](#)。
- IAM 執行個體描述檔，具有呼叫 AWS PCS RegisterComputeNodeGroupInstance API 動作和存取節點群組執行個體所需任何其他 AWS 資源的許可。如需詳細資訊，請參閱 [平行運算服務的 IAM AWS 執行個體描述檔](#)。
- 節點群組執行個體的啟動範本。如需詳細資訊，請參閱 [搭配 AWS PCS 使用 Amazon EC2 啟動範本](#)。
- 若要建立使用 Amazon EC2 Spot 執行個體的運算節點群組，您必須在您的 中具有 AWSServiceRoleForEC2Spot 服務連結角色 AWS 帳戶。如需詳細資訊，請參閱 [AWS PCS 的 Amazon EC2 Spot 角色](#)。

在 AWS PCS 中建立運算節點群組

您可以使用 AWS 管理主控台 或 建立運算節點群組 AWS CLI。

AWS 管理主控台

使用主控台建立運算節點群組

1. 開啟 [AWS PCS 主控台](#)。
2. 選取您要建立運算節點群組的叢集。導覽至運算節點群組，然後選擇建立。
3. 在運算節點群組設定區段中，提供節點群組的名稱。名稱只能包含區分大小寫的英數字元和連字號。它必須以字母字元開頭，且長度不可超過 25 個字元。名稱在叢集中必須是唯一的。
4. 在運算組態下，輸入或選取這些值：
 - a. EC2 啟動範本 – 選取要用於此節點群組的自訂啟動範本。啟動範本可用來自訂網路設定，例如子網路、安全群組、監控組態和執行個體層級儲存。如果您沒有準備好啟動範本，請參閱 [以搭配 AWS PCS 使用 Amazon EC2 啟動範本](#) 了解如何建立範本。

⚠ Important

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

- b. EC2 啟動範本版本 – 您必須選取自訂啟動範本的版本。如果您稍後變更版本，則必須更新運算節點群組，以偵測啟動範本中的變更。如需詳細資訊，請參閱[更新 AWS PCS 運算節點群組](#)。
 - c. AMI ID – 如果您的啟動範本不包含 AMI ID，或您想要覆寫啟動範本中的值，請在此處提供 AMI ID。請注意，用於節點群組的 AMI 必須與 AWS PCS 相容。您也可以選取提供的範例 AMI AWS。如需此主題的詳細資訊，請參閱[AWS PCS 的 Amazon Machine Image AMIs](#)。
 - d. IAM 執行個體描述檔 – 選擇節點群組的執行個體描述檔。執行個體描述檔會授予執行個體安全存取 AWS 資源和服務的權限。如果您沒有準備好，您可以選取建立基本設定檔，讓 AWS PCS 使用最低政策為您建立設定檔，或參閱[平行運算服務的 IAM AWS 執行個體描述檔](#)。
 - e. 子網路 – 在部署 AWS PCS 叢集的 VPC 中選擇一或多個子網路。如果您選擇多個子網路，則節點之間無法使用 EFA 通訊，而不同子網路中節點之間的通訊可能會增加延遲。請確定您在此處指定的子網路符合您在 EC2 啟動範本中定義的任何子網路。
 - f. 執行個體 – 選擇一或多個執行個體類型，以滿足節點群組中的擴展請求。所有執行個體類型都必須具有相同的處理器架構 (x86_64 或 arm64) 和 vCPUs 數量。如果執行個體具有 GPUs，則所有執行個體類型都必須具有相同數目的 GPUs。
 - g. 擴展組態 – 指定節點群組的執行個體數量下限和上限。您可以定義靜態組態，其中有固定數量的節點正在執行，或是動態組態，其中最多可以執行節點的計數上限。對於靜態組態，將最小值和最大值設定為相同，大於零個數字。對於動態組態，請將最小執行個體設定為零，最大執行個體設定為大於零的數字。AWS PCS 不支援混合靜態和動態執行個體的運算節點群組。
5. (選用) 在其他設定下，指定下列項目：
- a. 購買選項 – 選取隨需執行個體、Spot 執行個體或現有的容量區塊。如果您計劃使用隨需容量保留 (ODCR)，也請選擇隨需。如需詳細資訊，請參閱[搭配 AWS PCS 使用 ODCRs](#)。選擇容量區塊，將現有的 Amazon EC2 容量區塊用於 ML 保留。如需詳細資訊，請參閱[搭配 AWS PCS 使用適用於 ML 的 Amazon EC2 容量區塊](#)。

- b. 配置策略 – 如果您已選取 Spot 購買選項，您可以指定在節點群組中啟動執行個體時如何選擇 Spot 容量集區。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 [Spot 執行個體的配置策略](#)。如果您已選取隨需購買選項，則此選項沒有作用。
6. (選用) 在 Slurm 自訂設定區段中，您可以新增參數名稱和值對來設定其他 Slurm 設定。如需支援參數的完整清單，請參閱 [AWS PCS 運算節點群組的自訂 Slurm 設定](#)。
7. (選用) 在標籤下，將任何標籤新增至運算節點群組。
8. 選擇建立運算節點群組。當 AWS PCS 佈建節點群組 Creating 時，狀態欄位會顯示。這可能需要幾分鐘的時間。

建議的下一個步驟

- 將節點群組新增至 AWS PCS 中的佇列，使其能夠處理任務。

AWS CLI

使用 建立運算節點群組 AWS CLI

使用下列命令建立佇列。執行命令之前，請執行下列替換：

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

Important

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

6. 將 `launch-template-version` 取代為特定的啟動範本版本。AWS PCS 會將您的節點群組與該特定版本的啟動範本建立關聯。

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

```
aws pcs create-compute-node-group --region region \
  --cluster-identifier my-cluster \
  --compute-node-group-name my-node-group \
  --subnet-ids subnet-ExampleID1 \
  --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
  --iam-instance-profile-arn=arn:InstanceProfile \
  --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
  --instance-configs instanceType=t3.large
```

Example– 使用自訂 Slurm 設定建立運算節點群組

```
aws pcs create-compute-node-group --region region \
  --cluster-identifier my-cluster \
  --compute-node-group-name my-node-group \
  --subnet-ids subnet-ExampleID1 \
  --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
  --iam-instance-profile-arn=arn:InstanceProfile \
  --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
  --instance-configs instanceType=t3.large \
  --slurm-configuration \
  'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

如需詳細資訊，請參閱 [AWS PCS 運算節點群組的自訂 Slurm 設定](#)。

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

- 您可以指定自訂啟動範本 `--amiId` 是否不包含 AMI 的參考，或是您想要覆寫該值。請注意，用於節點群組的 AMI 必須與 AWS PCS 相容。您也可以選取提供的範例 AMI AWS。如需此主題的詳細資訊，請參閱 [AWS PCS 的 Amazon Machine Image AMIs](#)。
- 使用 `--purchase-option` 選擇 AWS PCS 為您的運算節點群組購買 EC2 執行個體的方式。隨需是預設值。
 - ONDEMAND – 使用隨需執行個體。如果您計劃使用隨需容量保留 (ODCR)，也請選擇此選項。如需詳細資訊，請參閱 [搭配 AWS PCS 使用 ODCRs](#)。
 - SPOT – 使用 Spot 執行個體。如果您選擇 Spot 執行個體，您也可以使用 `--allocation-strategy` 定義 AWS PCS 在節點群組中啟動執行個體時如何選擇 Spot 容量集區。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 [Spot 執行個體的配置策略](#)。
 - CAPACITY_BLOCK – 使用現有的 Amazon EC2 容量區塊進行 ML 保留。如需詳細資訊，請參閱 [搭配 AWS PCS 使用適用於 ML 的 Amazon EC2 容量區塊](#)。
- 您可以使用為節點群組中的節點提供 Slurm 組態選項 `--slurm-configuration`。您可以設定權重（排程優先順序）和實際記憶體。權重較低的節點具有較高的優先順序，而且單位是任意的。如需詳細資訊，請參閱 Slurm 文件中的 [權重](#)。實際記憶體會設定節點群組中節點上實際記憶體的大小（以 GB 為單位）。其旨在與 Slurm 組態中 AWS PCS 中叢集的 `CR_CPU_Memory` 選項搭配使用。如需詳細資訊，請參閱 Slurm 文件中的 [RealMemory](#)。

Important

建立運算節點群組可能需要幾分鐘的時間。

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

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

更新 AWS PCS 運算節點群組

本主題提供可用選項的概觀，並說明更新 AWS PCS 運算節點群組時的考量事項。如需 Slurm 自訂設定的詳細資訊，請參閱 [AWS PCS 運算節點群組的自訂 Slurm 設定](#)。

更新 AWS PCS 運算節點群組的選項

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

下列節點群組設定無法在建立後變更：

- 名稱
- 執行個體

更新 AWS PCS 運算節點群組時的考量事項

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

- 當運算節點群組狀態從更新到作用中時，運算節點群組屬性的變更就會生效。使用更新的屬性啟動新的執行個體。
- 不會影響特定節點組態的更新不會影響執行中的節點。例如，新增子網路並變更配置策略。
- 如果您更新運算節點群組的啟動範本，則必須更新運算節點群組以使用新版本。
- 若要從運算節點群組中的節點新增或移除安全群組，請編輯其啟動範本並更新運算節點群組。使用更新的一組安全群組啟動新的執行個體。
- 如果您直接編輯運算節點群組使用的安全群組，它會立即影響執行中和未來的執行個體。
- 如果您從運算節點群組使用的 IAM 執行個體描述檔新增或移除許可，它會立即對執行中和未來的執行個體生效。
- 若要變更運算節點群組執行個體使用的 AMI，請更新運算節點群組（或其啟動範本）以使用新的 AMI，並等待 AWS PCS 取代執行個體。
- AWS PCS 會在節點群組更新操作後取代節點群組中的現有執行個體。如果有任務在節點上執行，則允許在 AWS PCS 取代節點之前完成這些任務。互動式使用者程序（例如登入節點執行個體）會終止。當 AWS PCS 標記執行個體進行取代Active時，節點群組狀態會傳回 `Updating`，但實際取代會在執行個體閒置時發生。
- 如果您減少運算節點群組中允許的執行個體數目上限，AWS PCS 會從 Slurm 移除節點，以符合新的上限。AWS PCS 會終止與已移除 Slurm 節點相關聯的執行中執行個體。已移除節點上的執行中任務會失敗並返回其佇列。

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

更新 AWS PCS 運算節點群組

您可以使用 AWS 管理主控台或 AWS CLI 更新節點群組。

AWS 管理主控台

更新運算節點群組

1. 在開啟 AWS PCS 主控台 <https://console.aws.amazon.com/pcs/home#/clusters>
2. 選取您要更新運算節點群組的叢集。
3. 導覽至運算節點群組，前往您要更新的節點群組，然後選取編輯。
4. 在運算組態、其他設定和Slurm自訂設定區段中，更新下列任何值：
 - 執行個體 – 您無法變更運算節點群組中的執行個體。

如需 Slurm 自訂設定的詳細資訊，請參閱 [AWS PCS 運算節點群組的自訂 Slurm 設定](#)。

5. 選擇更新。套用變更時，狀態欄位會顯示更新。

Important

運算節點群組更新可能需要幾分鐘的時間。

AWS CLI

更新運算節點群組

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

- c. 以叢集的名稱或 取代 *my-cluster*。 clusterId

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

Example– 使用自訂 Slurm 設定更新運算節點群組

```
aws pcs update-compute-node-group --region region-code \  
  --cluster-identifier my-cluster \  
  --compute-node-group-identifier my-node-group \  
  --slurm-configuration \  
  'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu, nvme"}]'
```

如需詳細資訊，請參閱[AWS PCS 運算節點群組的自訂 Slurm 設定](#)。

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

Important

更新運算節點群組可能需要幾分鐘的時間。

您可以使用下列命令查詢節點群組的狀態。

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

在 AWS PCS 中刪除運算節點群組

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

刪除運算節點群組時的考量事項

運算節點群組定義用於處理任務、提供互動式 shell 存取和其他任務的 EC2 執行個體。它們通常與一個或多個 AWS PCS 佇列相關聯。刪除運算節點群組之前，請考慮下列事項：

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

刪除運算節點群組

您可以使用 AWS 管理主控台 或 AWS CLI 來刪除運算節點群組。

AWS 管理主控台

刪除運算節點群組

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

Note

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

AWS CLI

刪除運算節點群組

- 使用下列命令來刪除具有這些替換項目的運算節點群組：
 - AWS 區域 將 *region-code* 取代為叢集所在的。

- 以運算節點 *my-node-group* 群組。
- 以叢集的名稱或 ID 取代 *my-cluster*。

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

刪除運算節點群組可能需要幾分鐘的時間。

Note

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

在 AWS PCS 中取得運算節點群組詳細資訊

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

AWS 管理主控台

若要取得運算節點群組詳細資訊

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

AWS CLI

若要取得運算節點群組詳細資訊

1. 使用 [ListClusters](#) API 動作來尋找您的叢集名稱或 ID。

```
aws pcs list-clusters
```

輸出範例：

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

2. 使用 [ListComputeNodeGroups](#) API 動作列出叢集中的運算節點群組。

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

呼叫範例：

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

輸出範例：

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

```

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

```

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

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

呼叫範例：

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

輸出範例：

```

{
  "computeNodeGroup": {
    "name": "compute-1",
    "id": "pcs_abc123abc1",
    "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
    computenodegroup/pcs_abc123abc1",
    "clusterId": "pcs_abc1234567",
    "createdAt": "2025-04-01T20:19:25+00:00",
    "modifiedAt": "2025-04-01T20:19:25+00:00",
    "status": "ACTIVE",
    "amiId": "ami-0123456789abcdef0",
    "subnetIds": [
      "subnet-abc012345789abc12"
    ],
    "purchaseOption": "ONDEMAND",
    "customLaunchTemplate": {
      "id": "lt-012345abcdef01234",
      "version": "1"
    }
  }
}

```

```
    },
    "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-profile/
AWSPCS-get-started-cfn-us-east-1",
    "scalingConfiguration": {
      "minInstanceCount": 0,
      "maxInstanceCount": 4
    },
    "instanceConfigs": [
      {
        "instanceType": "c6i.xlarge"
      }
    ]
  }
}
```

尋找 AWS PCS 中的運算節點群組執行個體

每個 AWS PCS 運算節點群組都可以使用共用組態啟動 EC2 執行個體。您可以使用 EC2 標籤，在 [AWS 管理主控台](#) 的運算節點群組中尋找執行個體 AWS CLI。

AWS 管理主控台

尋找您的運算節點群組執行個體

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

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

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

AWS CLI

若要尋找節點群組執行個體，請使用以下命令。執行命令之前，請進行下列取代：

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

```
aws ec2 describe-instances \
  --region region-code --filters \
  "Name=tag:aws:pcs:compute-node-group-id,Values=node-group-id" \
  "Name=instance-state-name,Values=running" \
  --query 'Reservations[*].Instances[*]'.
[{InstanceID:InstanceId,State:State.Name,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress}]
```

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

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

Note

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

搭配 AWS PCS 使用 Amazon EC2 啟動範本

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

了解撰寫自訂啟動範本時有哪些選項和考量事項，可協助您撰寫用於 AWS PCS 的選項和考量事項。如需啟動範本的詳細資訊，請參閱《Amazon EC2 使用者指南》中的[從啟動範本啟動執行個體](#)。

主題

- [AWS PCS 啟動範本概觀](#)
- [建立基本的啟動範本](#)
- [使用 AWS PCS 的 Amazon EC2 使用者資料](#)
- [AWS PCS 中的容量保留](#)
- [有用的啟動範本參數](#)

AWS PCS 啟動範本概觀

您可以在 EC2 啟動範本中包含[超過 30 個參數](#)，控制執行個體設定方式的許多層面。大多數都與 AWS PCS 完全相容，但有一些例外狀況。

AWS PCS 會忽略 EC2 啟動範本的下列參數，因為這些屬性必須由服務直接管理：

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

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

這些參數具有支援 AWS PCS 相容性的特殊需求：

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

建立基本的啟動範本

您可以使用 AWS 管理主控台 或 建立啟動範本 AWS CLI。

AWS 管理主控台

建立啟動範本

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

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

建議的下一個步驟

- 使用新的啟動範本來建立或更新 AWS PCS 運算節點群組。

AWS CLI

建立啟動範本

使用下列命令建立您的啟動範本。

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

```
aws ec2 create-launch-template --region region-code \  
  --launch-template-name my-template-name \  
  --launch-template-data '{"KeyName":"my-ssh-key-name","SecurityGroupIds":  
  ["sg-ExampleID1","sg-ExampleID2"]}'
```

AWS CLI 將輸出類似以下內容的文字。啟動範本 ID 可在 中找到LaunchTemplateId。

```
{  
  "LaunchTemplate": {  
    "LatestVersionNumber": 1,  
    "LaunchTemplateId": "lt-0123456789abcdef01",  
    "LaunchTemplateName": "my-launch-template-name",  
    "DefaultVersionNumber": 1,  
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",  
    "CreateTime": "2019-04-30T18:16:06.000Z"  
  }  
}
```

建議的下一個步驟

- 使用新的啟動範本來建立或更新 AWS PCS 運算節點群組。

使用 AWS PCS 的 Amazon EC2 使用者資料

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

我們的使用者資料可以執行常見的組態案例，包括但不限於下列項目：

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

啟動範本中的使用者資料必須採用 [MIME 分段封存](#) 格式。這是因為您的使用者資料會與節點群組中設定節點所需的其他 AWS PCS 使用者資料合併。您可以將多個使用者資料區塊組合在一起成為單一 MIME 分段檔案。

MIME 分段檔案包含下列元件：

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

Note

如果您在 Amazon EC2 主控台中將使用者資料新增至啟動範本，您可以將其貼上為純文字。或者，您可以從檔案上傳。如果您使用 AWS CLI 或 AWS 開發套件，您必須先對使用者資料進行 base64 編碼，並在呼叫 [CreateLaunchTemplate](#) 時提交該字串做為 UserData 參數的值，如此 JSON 檔案所示。

```
{
  "LaunchTemplateName": "base64-user-data",
  "LaunchTemplateData": {
    "UserData":
    "ewogICAgIkkxhdW5jaFRlbXBsYXR1TmFtZSI6ICJpbmNyZWZzZS1jb250YWluZXItZm9sdW..."
  }
}
```

範例

- [範例：從套件儲存庫安裝軟體](#)
- [範例：從 S3 儲存貯體執行指令碼](#)
- [範例：設定全域環境變數](#)
- [搭配 AWS PCS 使用網路檔案系統](#)
- [範例：使用 EFS 檔案系統做為共用主目錄](#)

範例：從套件儲存庫安裝 AWS PCS 的軟體

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

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

Note

您的執行個體必須能夠連線到其設定的套件儲存庫。

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

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

packages:
- python3-devel
- rust
- golang

--===MYBOUNDARY===
```

範例：從 S3 儲存貯體執行 AWS PCS 的其他指令碼

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

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

將下列值取代為您自己的詳細資訊：

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

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

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

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/object-key /tmp/script.sh
- /usr/bin/shell /tmp/script.sh

--===MYBOUNDARY===--
```

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

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
```

```

        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
}

```

範例：設定 AWS PCS 的全域環境變數

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

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

```

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

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
touch /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR1=100 >> /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR2=abc >> /etc/profile.d/awspcs-userdata-vars.sh

--==MYBOUNDARY==--

```

範例：使用 EFS 檔案系統做為 AWS PCS 的共用主目錄

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

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

使用您自己的詳細資訊取代此指令碼中的下列值：

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

```

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

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

packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /tmp/home
  - rsync -a /home/ /tmp/home
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults
  - rsync -a --ignore-existing /tmp/home/ /home
  - rm -rf /tmp/home/

--==MYBOUNDARY==--

```

範例：啟用無密碼 SSH

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

```

#!/bin/bash

mkdir -p $HOME/.ssh && chmod 700 $HOME/.ssh
touch $HOME/.ssh/authorized_keys
chmod 600 $HOME/.ssh/authorized_keys

if [ ! -f "$HOME/.ssh/id_rsa" ]; then
    ssh-keygen -t rsa -b 4096 -f $HOME/.ssh/id_rsa -N ""
    cat ~/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
fi

```

Note

執行個體必須使用允許叢集節點之間 SSH 連線的安全群組。

AWS PCS 中的容量保留

您可以使用 ML 的隨需容量預留或 Amazon EC2 容量區塊，在特定的持續時間內保留特定可用區域中的 Amazon EC2 容量，以確保您在需要時擁有可用的必要運算容量。

隨需容量保留 (ODCRs) 可讓您為特定可用區域中的 Amazon EC2 執行個體保留任何持續時間的運算容量。您可以隨時建立和取消保留，無需長期承諾或預付款項。當您需要可隨需求變更而修改的彈性容量保留時，ODCRs 是理想的選擇。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 [隨需容量保留](#)。

Amazon EC2 Capacity Blocks for ML 可讓您預留 GPU 加速運算執行個體以供日後使用，最多可提前 8 週。您可以保留 1-64 個執行個體的區塊，持續時間從 1 天到 6 個月。容量區塊非常適合需要在特定時間保證存取 GPU 容量的機器學習工作負載。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 [ML 容量區塊](#)。

主題

- [搭配 AWS PCS 使用 ODCRs](#)
- [搭配 AWS PCS 使用適用於 ML 的 Amazon EC2 容量區塊](#)

搭配 AWS PCS 使用 ODCRs

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

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

1. 使用 [Amazon EC2 建立容量保留使用者指南 建立目標隨需容量保留 \(ODCR\)](#)。
2. 將 ODCR 與啟動範本建立關聯。有兩種方法可以執行此操作：
 - a. 直接 ODCR 關聯：直接在啟動範本中參考 ODCR ID。此方法提供嚴格的容量控制，並不支援執行個體回填（如果運算節點群組請求的執行個體超過 ODCR 中可用的執行個體，則不會啟動其他執行個體）。
 - b. 容量保留群組關聯：將 ODCR 新增至容量保留群組，並在啟動範本中參考群組。此方法支援執行個體回填，允許 AWS PCS 在超過保留容量時啟動額外的隨需執行個體。
3. 建立或更新 AWS PCS 運算節點群組以使用啟動範本。如需詳細資訊，請參閱 [AWS PCS 運算節點群組使用者指南](#)。

- 將運算節點群組purchaseOption的 設定為 ONDEMAND。

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

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

```
aws ec2 create-capacity-reservation \  
  --instance-type hpc6a.48xlarge \  
  --instance-platform Linux/UNIX \  
  --availability-zone us-east-2a \  
  --instance-count 32 \  
  --instance-match-criteria targeted
```

此命令的結果將是新 ODCR 的 ARN。您可以從 ARN `arn:aws:ec2:us-east-2:123456789012:capacity-reservation/ODCR-ID` 或使用 [Amazon EC2 DescribeCapacityReservations](#) 擷取 ODCR ID。

直接 ODCR 關聯：將 ODCR ID 新增至啟動範本。以下是參考 ODCR ID 的範例啟動範本。

```
{  
  "CapacityReservationSpecification": {  
    "CapacityReservationTarget": {  
      "CapacityReservationId": "cr-1234567890abcdef1"  
    }  
  }  
}
```

容量保留群組關聯：建立容量保留群組，並將群組新增至啟動範本。下列命令會建立名為 的容量保留群組 `EXAMPLE-CR-GROUP`。

```
aws resource-groups create-group \  
  --name EXAMPLE-CR-GROUP \  
  --configuration \  
    '{"Type": "AWS::EC2::CapacityReservationPool"}' \  
    '{"Type": "AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]]'
```

下列命令會將 ODCR 新增至容量保留群組。

```
aws resource-groups group-resources --group EXAMPLE-CR-GROUP \  
  --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/  
cr-1234567890abcdef1
```

建立 ODCR 並新增至容量保留群組後，現在可以透過將 ODCR 新增至啟動範本來連線至 AWS PCS 運算節點群組。以下是參考容量保留群組的範例啟動範本。

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

最後，建立或更新 AWS PCS 運算節點群組以使用 hpc6a.48xlarge 執行個體，並使用參考 ODCR 的啟動範本。對於靜態節點群組，請將最小和最大執行個體設定為保留的大小 (32)。對於動態節點群組，將最小執行個體設定為 0，並將最大執行個體設定為所需的執行個體大小。

此範例是為一個運算節點群組佈建的單一 ODCR 的簡單實作。但是，AWS PCS 支援許多其他設計。例如，您可以在多個運算節點群組之間分割大型 ODCR 或容量保留群組。或者，您可以使用另一個 AWS 帳戶已建立並與您共用的 ODCRs。

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

搭配 AWS PCS 使用適用於 ML 的 Amazon EC2 容量區塊

Amazon EC2 Capacity Blocks for ML 是一種 Amazon EC2 購買選項，可讓您預先付費，在特定日期和時間範圍內保留 GPU 加速運算執行個體，以支援短期工作負載。容量區塊內執行的執行個體會在 Amazon EC2 UltraCluster 內自動放置於鄰近位置，以實現低延遲、Pb 級的非阻塞式聯網。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 [ML 的容量區塊](#)。

您可以使用啟動範本，讓 AWS PCS 在為運算節點群組啟動執行個體時，使用容量區塊。

Note

AWS PCS 自 Slurm 24.05 版開始推出對容量區塊的支援。

限制

- AWS PCS 僅支援具有 P6-B300, P6-B200, P5en, P5e, P5 和 P4d 執行個體系列的容量區塊。
- 您一次只能將運算節點群組與 1 個容量區塊建立關聯。
- 您無法將運算節點群組與結合多個容量區塊的容量保留群組建立關聯。
- 容量區塊必須處於 `scheduled` 或 `active` 狀態，才能與 AWS PCS 搭配使用。您無法在其他狀態下使用容量區塊，例如 `payment-failed`。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的[檢視容量區塊](#)。
- 對於 P6 和 P5 執行個體類型，請參閱相關的 AWS 文件：[P6 執行個體的軟體需求](#)、[使用多個網路卡最大化 Amazon EC2 執行個體的網路頻寬](#)

容量區塊過期

容量區塊僅限於特定日期和時間範圍。當容量區塊過期時：

- 與該容量區塊相關聯的運算節點群組會繼續存在，並保持與相同佇列的關聯。
- 運算節點群組中的所有執行個體都會終止，且作用中任務可能會根據您的 Slurm 設定失敗。
- AWS PCS 無法在運算節點群組中啟動新的執行個體。
- 所有佇列或新提交的任務都會保持待定狀態，直到另一個運算節點群組連接到佇列，或者您更新運算節點群組以使用指定新容量區塊的新啟動範本。

設定 AWS PCS 運算節點群組以使用容量區塊

將容量區塊與運算節點群組建立關聯

1. 為 AWS PCS 建立指定容量區塊的 Amazon EC2 啟動範本。如需為 AWS PCS 建立啟動範本的詳細資訊，請參閱[搭配 AWS PCS 使用 Amazon EC2 啟動範本](#)。

您的啟動範本必須包含：

- 的值 `MarketTypeInstanceMarketOptions` 必須設定為 `capacity-block`。
 - `CapacityReservationSpecification` 具有有效的 `CapacityReservationId`
 - `InstanceType` 與您購買的容量區塊的執行個體類型相符的有效。
2. 建立使用啟動範本的運算節點群組。如需詳細資訊，請參閱[在 AWS PCS 中建立運算節點群組](#)。您也可以更新現有的運算節點群組，以使用啟動範本。如需詳細資訊，請參閱[更新 AWS PCS 運算節點群組](#)。

當您建立或更新運算節點群組時：

- 您用來建立或更新運算節點群組的 IAM 身分必須具有下列許可：

```
ec2:DescribeCapacityReservations
```

如需詳細資訊，請參閱[AWS PCS 的最低許可](#)。

- 容量區塊必須處於 `scheduled` 或 `active` 狀態。
- 將運算節點群組 `purchaseOption` 的設定為 `CAPACITY_BLOCK`。
- 運算節點群組 `maxInstanceCount` 的不得超過容量區塊的大小。
- 運算節點群組的可用區域必須符合運算節點群組子網路可用區域的 1 個。

Important

您無法在更新運算節點群組時變更其執行個體類型。您只能使用與運算節點群組具有相同執行個體類型的容量區塊。如果您想要使用具有不同執行個體類型的容量區塊，則必須建立新的運算節點群組。

有關搭配 AWS PCS 使用容量區塊的常見問題

我剛支付容量區塊的費用，並立即嘗試將其與 AWS PCS 搭配使用，但運算節點群組建立失敗。發生了什麼？

您的容量區塊可能不是 `scheduled` 或 `active` 狀態。請在容量區塊為 `scheduled` 或之後再試一次 `active`。

我在 AWS PCS 中使用容量區塊，並在過期之前購買擴充功能。如何在 AWS PCS 中繼續使用它？

您不需要採取任何動作即可繼續使用 AWS PCS 中的容量區塊。容量區塊的結束日期會在您的延伸付款成功後更新。只要容量區塊沒有過期，運算節點群組就會繼續運作。如果您的延伸付款失敗，您的容量區塊仍會保留，`active` 且運算節點群組會持續運作，直到容量區塊在原始結束日期過期為止。

如果我的容量區塊過期，佇列和執行中的任務會發生什麼情況？

在容量區塊過期之前未啟動的佇列任務會保持待定狀態，直到您將另一個運算節點群組連接到佇列，或您使用新的容量區塊更新運算節點群組為止。您仍然可以將任務提交至佇列。您的 Slurm 設定會影響作用中的任務。根據預設，作用中的任務會自動重新排入佇列，但可能會有錯誤或失敗。

我的容量區塊已過期。我應該做些事嗎？

您不需要執行任何動作。您可以檢查 Amazon EC2 主控台的 EC2 容量保留狀態。當容量區塊過期時，與該容量區塊相關聯的運算節點群組會繼續存在並處理相同的佇列。運算節點群組沒有任何執行個體可執行任務。您可以刪除運算節點群組，或取消其與佇列的關聯，以防止使用者提交無法執行的任務。

我想要在 AWS PCS 運算節點群組中使用新的容量區塊。我該怎麼辦？

我們建議您建立新的運算節點群組，以使用新的容量區塊。如需詳細資訊，請參閱[設定 AWS PCS 運算節點群組以使用容量區塊](#)。

如何跨叢集和服務共用 1 個容量區塊？

您可以將容量區塊分割到多個叢集和服務。例如，若要將容量區塊分割為 PCS-Cluster-1 上有 20 個節點的 64 個 p5.48xlarge 執行個體、PCS-Cluster-2 有 16 個節點，以及其他服務的其餘節點，請將 PCS-Cluster-1 maxInstanceCount minInstanceCount 和 PCS-Cluster-2 的和 16 個節點設為 20。

我可以搭配 1 個運算節點群組使用多個容量區塊或合併容量嗎？

否。只有 1 個容量區塊可以與單一運算節點群組相關聯。AWS PCS 不支援結合多個容量區塊的容量保留群組。

如何知道我的容量區塊何時開始或過期？

與 AWS PCS 無關，Amazon EC2 會在容量區塊保留開始時透過 EventBridge 傳送 Capacity Block Reservation Delivered 事件，並在容量區塊保留到期前 40 分鐘傳送 Capacity Block Reservation Expiration Warning 事件。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的[使用 EventBridge 監控容量區塊](#)。

Slurm 如何追蹤容量區塊的狀態？

您可以執行 `sinfo` 以了解 AWS PCS 如何使用容量區塊。在下列範例輸出中，佇列與從 active 容量區塊執行 4 個執行個體的運算節點群組相關聯。節點處於 idle Slurm 狀態（可供使用且尚未配置給任何任務）。

```
$ sinfo
```

```
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
fanout up infinite 4 idle node-fanout-[1-4]
```

如果節點處於 `maint` 狀態，您可以執行 `scontrol show res` 以查看控制此狀態的 Slurm 保留的詳細資訊。在下列範例輸出中，容量區塊 `scheduled` 具有未來的開始日期。

```
$ scontrol show res

ReservationName=node-fanout-scheduled StartTime=2025-10-14T13:09:17
EndTime=2025-10-14T13:11:17 Duration=00:02:00
  Nodes=node-fanout-[1-4] NodeCnt=4 CoreCnt=16 Features=(null) PartitionName=(null)
Flags=MAINT,SPEC_NODES
  TRES=cpu=16

  Users=root Groups=(null) Accounts=(null) Licenses=(null) State=ACTIVE
BurstBuffer=(null)
  MaxStartDelay=(null)

Comment=node-fanout Scheduled
```

如何判斷我在啟動容量時遇到的錯誤是否因為共用容量區塊而發生？

在 Amazon EC2 主控台中檢查容量保留，以尋找容量區塊中主動佈建的執行個體數量。檢查每個執行個體的標籤，以尋找哪些服務或叢集使用它。例如，AWS PCS 的所有執行個體都有 AWS PCS 標籤 `aws:pcs:cluster-id = pcs_l0mizqyk5o | aws:pcs:compute-node-group-id = pcs_ic7onkmfqk`，例如指出執行個體所屬的叢集和運算節點群組。然後，您可以檢查容量區塊是否達到容量上限。

您可以使用 `scontrol show nodes` 來檢查 AWS PCS 叢集中的容量區塊節點是否正在觸發 `ReservationCapacityExceeded`：

```
[root@ip-172-16-10-54 ~]# scontrol show nodes test-node-8-gamma-cb-2
NodeName=test-8-gamma-cb-2 CoresPerSocket=1
  CPUAlloc=0 CPUEfctv=8 CPUTot=8 CPULoad=0.00
  AvailableFeatures=test-8-gamma-cb,gpu
  ActiveFeatures=test-8-gamma-cb,gpu
  Gres=gpu:H100:1
  NodeAddr=test-8-gamma-cb-2 NodeHostName=test-8-gamma-cb-2
  RealMemory=249036 AllocMem=0 FreeMem=N/A Sockets=8 Boards=1
  State=IDLE+CLOUD+POWERING_DOWN ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A
MCS_label=N/A
```

```
Partitions=my-q
BootTime=None SlurmdStartTime=None
LastBusyTime=Unknown ResumeAfterTime=None
CfgTRES=cpu=8,mem=249036M,billing=8
AllocTRES=
CurrentWatts=0 AveWatts=0
Reason=Failed to launch backing instance (Error Code:
ReservationCapacityExceeded) [root@2025-08-28T15:15:33]
```

當多個運算節點群組連接到相同的佇列時，如何強制任務在容量區塊支援的執行個體上執行？

您可以使用 Slurm 功能和限制條件，將任務鎖定到特定節點集。我們建議您不要為每個運算節點群組設定 Slurm 權重，因為僅適用於未處於 maint 狀態的節點。

有用的啟動範本參數

本節說明一些可能對 AWS PCS 廣泛有用的啟動範本參數。

開啟詳細的 CloudWatch 監控

您可以使用啟動範本參數，以較短的時間隔啟用 CloudWatch 指標的集合。

AWS 管理主控台

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

YAML

```
Monitoring:
  Enabled: True
```

JSON

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

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

執行個體中繼資料服務第 2 版 (IMDS v2)

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

AWS 管理主控台

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

YAML

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

JSON

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

AWS PCS 佇列

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

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

主題

- [在 AWS PCS 中建立佇列](#)
- [更新 AWS PCS 佇列](#)
- [在 AWS PCS 中刪除佇列](#)

在 AWS PCS 中建立佇列

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

Note

您可以在佇列上設定自訂 Slurm 設定，以實作分割區特定的排程政策和資源管理。如需詳細資訊，請參閱[在 AWS PCS 中設定自訂 Slurm 設定](#)。

先決條件

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

在 AWS PCS 中建立佇列

您可以使用 AWS 管理主控台 或 建立佇列 AWS CLI。

AWS 管理主控台

使用主控台建立佇列

1. 開啟 [AWS PCS 主控台](#)。
2. 選取佇列的叢集。導覽至佇列，然後選擇建立佇列。
3. 在佇列組態區段中，提供下列值：
 - a. 佇列名稱 – 佇列的名稱。此名稱僅能使用英數字元 (區分大小寫) 和連字號。它必須以字母字元開頭，且長度不可超過 25 個字元。名稱在叢集中必須是唯一的。
 - b. 運算節點群組 – 選取 1 個或多個運算節點群組來服務此佇列。運算節點群組可以與多個佇列建立關聯。
4. (選用) 在其他排程器設定區段中，您可以新增參數名稱和值對來設定其他 Slurm 設定。如需支援參數的完整清單，請參閱 [AWS PCS 佇列的自訂 Slurm 設定](#)。
5. (選用) 在標籤下，將任何標籤新增至 AWS PCS 佇列
6. 選擇建立佇列。當 AWS PCS 建立佇列時，狀態欄位會顯示建立。建立佇列可能需要幾分鐘的時間。

建議的下一個步驟

- 將任務提交到您的新佇列。

AWS CLI

使用 建立佇列 AWS CLI

使用下列命令來建立佇列。進行下列取代：

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

Note

建立佇列時，您必須提供運算節點群組的 ID，而不是其名稱。

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

Example– 使用自訂 Slurm 設定建立佇列

```
aws pcs create-queue --region region-code \  
  --queue-name my-queue \  
  --cluster-identifier my-cluster \  
  --compute-node-group-configurations \  
  computeNodeGroupId=compute-node-group-id \  
  --slurm-configuration \  
  'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

如需詳細資訊，請參閱[AWS PCS 佇列的自訂 Slurm 設定](#)。

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

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

建議的下一個步驟

- 將任務提交到您的新佇列

更新 AWS PCS 佇列

本主題提供可用選項的概觀，並說明更新 AWS PCS 佇列時的考量事項。如需 Slurm 自訂設定的詳細資訊，請參閱[AWS PCS 佇列的自訂 Slurm 設定](#)。

更新 AWS PCS 佇列時的考量事項

佇列更新不會影響執行中的任務，但叢集在更新佇列時可能無法接受新任務。

更新 AWS PCS 佇列

您可以使用 AWS 管理主控台 或 AWS CLI 來更新佇列。

AWS 管理主控台

更新佇列

1. 在 開啟 AWS PCS 主控台 <https://console.aws.amazon.com/pcs/home#/clusters>
2. 選取您要更新佇列的叢集。
3. 導覽至佇列，前往要更新的佇列，然後選取編輯。
4. 在佇列組態區段中，更新下列任何值：
 - 節點群組 – 新增或移除運算節點群組與佇列的關聯。
 - 其他排程器設定 – 新增、修改或移除佇列的自訂 Slurm 設定。如需詳細資訊，請參閱 [AWS PCS 佇列的自訂 Slurm 設定](#)。
 - 標籤 – 新增或移除佇列的標籤。
5. 選擇更新。套用變更時，狀態欄位會顯示更新。

Important

佇列更新可能需要幾分鐘的時間。

AWS CLI

更新佇列

1. 使用下列命令更新您的佇列。執行命令之前，請執行下列替換：
 - a. 將 *region-code* 取代為您要在其中建立叢集 AWS 區域的。
 - b. 將 *my-queue* 取代 `computeNodeId` 為佇列的名稱或。
 - c. 以叢集的名稱或 取代 *my-cluster*。 `clusterId`

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

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

```
--compute-node-group-configurations
computeNodeId=computeNodeGroupExampleID1,computeNodeId=computeNodeGroupExampleID2
```

```
aws pcs update-queue --region region-code \
  --queue-identifier my-queue \
  --cluster-identifier my-cluster \
  --compute-node-group-configurations \
  computeNodeId=computeNodeGroupExampleID1
```

Example– 使用自訂 Slurm 設定更新佇列

```
aws pcs update-queue --region region-code \
  --queue-identifier my-queue \
  --cluster-identifier my-cluster \
  --slurm-configuration \
  'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

如需詳細資訊，請參閱 [AWS PCS 佇列的自訂 Slurm 設定](#)。

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

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

建議的後續步驟

- 將任務提交至更新後的佇列。

在 AWS PCS 中刪除佇列

本主題提供如何在 AWS PCS 中刪除佇列的概觀。

刪除佇列時的考量事項

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

刪除佇列

您可以使用 AWS 管理主控台 或 AWS CLI 來刪除佇列。

AWS 管理主控台

刪除佇列

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

Note

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


AWS CLI

刪除佇列

- 使用下列命令來刪除具有這些替換項目的佇列：
 - AWS 區域 將 *region-code* 取代為叢集所在的。
 - 以佇列的名稱或 ID 取代 *my-queue*。
 - 以叢集的名稱或 ID 取代 *my-cluster*。

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

刪除佇列可能需要幾分鐘的時間。

 Note

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

AWS PCS 登入節點

AWS PCS 叢集通常需要至少 1 個登入節點，以支援互動式存取和任務管理。達成此目的的一種方法是使用為登入節點功能設定的靜態 AWS PCS 運算節點群組。您也可以設定獨立 EC2 執行個體做為登入節點。

主題

- [使用 AWS PCS 運算節點群組提供登入節點](#)
- [使用獨立執行個體做為 AWS PCS 登入節點](#)
- [將獨立登入節點連接到 AWS PCS 中的多個叢集](#)

使用 AWS PCS 運算節點群組提供登入節點

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

建立登入節點的 AWS PCS 運算節點群組

在操作上，這與建立一般運算節點群組沒有太大不同。不過，有一些關鍵組態選擇：

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

- 在登入節點上設定與運算執行個體相同的網路檔案系統 (Amazon EFS、Amazon FSx for Lustre 等) 掛載。如需詳細資訊，請參閱[搭配 AWS PCS 使用網路檔案系統](#)。

存取您的登入節點

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

更新登入節點的 AWS PCS 運算節點群組

您可以使用 UpdateComputeNodeGroup 更新登入節點群組。作為節點群組更新程序的一部分，將取代執行中的執行個體。請注意，這會中斷執行個體上任何作用中的使用者工作階段或程序。執行中或排入佇列的 Slurm 任務將不受影響。如需詳細資訊，請參閱[更新 AWS PCS 運算節點群組](#)。

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

刪除登入節點的 AWS PCS 運算節點群組

您可以使用 AWS PCS 中的刪除運算節點群組機制來更新登入節點群組。執行中的執行個體將終止為節點群組刪除的一部分。請注意，這會中斷執行個體上任何作用中的使用者工作階段或程序。執行中或排入佇列的 Slurm 任務將不受影響。如需詳細資訊，請參閱[在 AWS PCS 中刪除運算節點群組](#)。

使用獨立執行個體做為 AWS PCS 登入節點

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

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

本教學課程可協助您設定連接至 AWS PCS 叢集的獨立執行個體。

內容

- [步驟 1 – 擷取目標 AWS PCS 叢集的地址和秘密](#)
- [步驟 2 – 啟動 EC2 執行個體](#)
- [步驟 3 – 在執行個體上安裝 Slurm](#)
- [步驟 4 – 擷取和存放叢集秘密](#)
- [步驟 5 – 設定 AWS PCS 叢集的連線](#)
- [步驟 6 – \(選用\) 測試連線](#)

步驟 1 – 擷取目標 AWS PCS 叢集的地址和秘密

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

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

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

命令會傳回類似此範例的輸出。

```
{
  "cluster": {
    "name": "get-started",
    "id": "pcs_123456abcd",
    "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
    "status": "ACTIVE",
    "createdAt": "2024-12-17T21:03:52+00:00",
    "modifiedAt": "2024-12-17T21:03:52+00:00",
    "scheduler": {
      "type": "SLURM",
      "version": "25.05"
    },
    "size": "SMALL",
    "slurmConfiguration": {
      "authKey": {
        "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!slurm-secret-pcs_123456abcd-a12ABC",
        "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
      }
    }
  }
}
```

```
    },
    "networking": {
      "subnetIds": [
        "subnet-0123456789abcdef0"
      ],
      "securityGroupIds": [
        "sg-0123456789abcdef0"
      ]
    },
    "endpoints": [
      {
        "type": "SLURMCTLD",
        "privateIpAddress": "10.3.149.220",
        "port": "6817"
      }
    ]
  }
}
```

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

步驟 2 – 啟動 EC2 執行個體

啟動 EC2 執行個體

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

- i. 選取 AWS PCS 叢集的 VPC。
- ii. 針對防火牆 (安全群組)，選擇選取現有的安全群組。
 - A. 選取允許執行個體與目標 AWS PCS 叢集 Slurm 控制器之間流量的安全群組。如需詳細資訊，請參閱[安全群組需求和考量事項](#)。
 - B. (選用) 選取允許傳入 SSH 存取執行個體的安全群組。
8. 在儲存區段中，視需要設定儲存磁碟區。請務必設定足夠的空間來安裝應用程式和程式庫，以啟用您的使用案例。
9. 在進階下，選擇允許存取叢集秘密的 IAM 角色。如需詳細資訊，請參閱[取得 Slurm 叢集秘密](#)。
10. 在摘要窗格中，選擇啟動執行個體。

步驟 3 – 在執行個體上安裝 Slurm

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

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

步驟 4 – 擷取和存放叢集秘密

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

使用下列命令存放叢集秘密。

- 建立 Slurm 的組態目錄。

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

- 擷取、解碼和存放叢集秘密。執行此命令之前，請將 `region-code` 取代為目標叢集正在執行的區域，並將 `secret-arn` 取代為[步驟 1](#)中 `secretArn` 擷取的值。

```
aws secretsmanager get-secret-value \  
  --region region-code \  
  --secret-id 'secret-arn' \  
  --version-stage AWSCURRENT \  
  --version-id secret-version-id
```

```
--query 'SecretString' \
--output text | base64 -d | sudo tee /etc/slurm/slurm.key
```

Warning

在多使用者環境中，任何可存取執行個體的使用者，如果可以存取執行個體中繼資料服務 (IMDS)，就可能可以擷取叢集秘密。這反過來可以讓他們模擬其他使用者。考慮將 IMDS 的存取權限制為僅限根使用者或管理使用者。或者，請考慮使用不依賴執行個體描述檔的不同機制來擷取和設定秘密。

- 設定 Slurm 金鑰檔案的擁有權和許可。

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

Note

Slurm 金鑰必須由執行 sackd 服務的使用者和群組所擁有。

步驟 5 – 設定 AWS PCS 叢集的連線

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

Note

如果您使用 Slurm 25.05 或更新版本，您可以使用指令碼來設定登入節點，以改為連線至多個叢集。如需詳細資訊，請參閱[將獨立登入節點連接到 AWS PCS 中的多個叢集](#)。

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

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

2. 建立用於管理 sackd 程序 systemd 的服務檔案。

```
sudo cat << EOF > /etc/systemd/system/sackd.service
```

```
[Unit]
Description=Slurm auth and cred kiosk daemon
After=network-online.target remote-fs.target
Wants=network-online.target
ConditionPathExists=/etc/sysconfig/sackd

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/sackd
User=slurm
Group=slurm
RuntimeDirectory=slurm
RuntimeDirectoryMode=0755
ExecStart=/opt/aws/pcs/scheduler/slurm-25.05/sbin/sackd --systemd \${SACKD_OPTIONS}
ExecReload=/bin/kill -HUP \${MAINPID}
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF
```

3. 設定sackd服務檔案的擁有權。

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

4. 啟用 sackd服務。

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

5. 啟動 sackd 服務。

```
sudo systemctl start sackd
```

步驟 6 – (選用) 測試連線

確認sackd服務正在執行。範例輸出如下。如果有錯誤，通常會顯示在這裡。

```
[root@ip-10-3-27-112 ~]# systemctl status sackd
```

```
[x] sackd.service - Slurm auth and cred kiosk daemon
   Loaded: loaded (/etc/systemd/system/sackd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-12-17 16:34:55 UTC; 8s ago
 Main PID: 9985 (sackd)
   CGroup: /system.slice/sackd.service
           ##9985 /opt/aws/pcs/scheduler/slurm-25.05/sbin/sackd --systemd --conf-
server=10.3.149.220:6817

Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Starting Slurm auth and cred
kiosk daemon...
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Started Slurm auth and cred
kiosk daemon.
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal sackd[9985]: sackd: running
```

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

```
[root@ip-10-3-27-112 ~]# /opt/aws/pcs/scheduler/slurm-25.05/bin/sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
all up infinite 4 idle~ compute-[1-4]
```

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

```
/opt/aws/pcs/scheduler/slurm-25.05/bin/srun --nodes=1 -p all --pty bash -i
```

將獨立登入節點連接到 AWS PCS 中的多個叢集

`pcs-multi-cluster-login-configure.sh` 指令碼提供在單一獨立登入節點上設定多個 Slurm `sackd` 協助程式的自動化方法。它可讓登入節點與多個叢集通訊。指令碼會自動執行下列操作：

- 使用 AWS PCS API 動作來取得叢集資訊
- base64 編碼 Slurm 身分驗證金鑰的提示
- 使用叢集身分驗證金鑰建立 Slurm JWKS 檔案
- 使用叢集端點和連接埠設定 `sackd` 服務
- 為叢集特定的 `sackd` 協助程式建立 `systemd` 服務檔案
- 產生叢集環境設定的啟用指令碼
- 啟用和啟動 `sackd` 服務

Note

此指令碼需要 Slurm 25.05 版或更新版本。

Slurm 必須已安裝在執行個體上（等同於手動程序中[的步驟 3](#)）。執行個體必須能夠到達目標叢集的端點。指令碼會執行手動組態程序中[步驟 4](#) 和 [步驟 5](#) 的同等操作。它會自動取得叢集資訊、設定 sckd 服務、建立必要的 systemd 服務檔案，以及建立啟用指令碼，讓使用者可用來設定其 shell 環境以進行叢集互動。

主題

- [AWS PCS 多叢集登入節點組態指令碼的先決條件](#)
- [AWS PCS 多叢集登入節點組態指令碼程式碼](#)
- [使用 AWS PCS 多叢集登入節點組態指令碼](#)

AWS PCS 多叢集登入節點組態指令碼的先決條件

系統要求

- systemd 支援 Linux 作業系統
- 系統組態的根權限

必要的命令和套件

- bash – Shell 解譯器 (4.0 版以上)
- curl – 適用於 AWS IMDS v2 中繼資料擷取
- jq – 用於剖析 AWS API 回應的 JSON 處理器
- aws – AWS CLI v2 執行 AWS PCS API 動作和 Secrets Manager 存取
- systemctl – systemd 服務管理
- find – 檔案系統搜尋公用程式
- grep – 文字模式比對
- sed – 用於文字處理的串流編輯器
- sort – 文字排序公用程式
- tail – 顯示檔案的最後一行

- `mkdir` – 建立目錄
- `chmod` – 變更檔案許可
- `chown` – 變更檔案擁有權
- `ldconfig` – 動態連結器組態

AWS 需求

- 執行 Slurm 25.05 版或更新版本的 AWS PCS 叢集
- AWS 設定的登入資料 (透過 IAM 角色、登入資料檔案或環境變數)
- 的許可：
 - `pcs:GetCluster`
 - `secretsmanager:GetSecretValue` (如果您使用替代秘密)

系統使用者和群組

- `slurm` 使用者和群組必須存在於系統上

Slurm 安裝

- Slurm 必須安裝在與 AWS PCS Slurm 安裝程式套件相同的位置：

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

AWS PCS 多叢集登入節點組態指令碼程式碼

將下列原始程式碼儲存至具有下列名稱的檔案：

```
pcs-multi-cluster-login-configure.sh
```

指令碼來源碼

```
#!/bin/bash
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# AWS PCS Multi-Cluster Standalone Login Node Configuration Script
```

```
#
# This script configures AWS Parallel Computing Service (PCS) multi-cluster stand alone
  login nodes
# by setting up the Slurm authentication and credential kiosk daemon (sackd)
# for connecting to remote PCS clusters.
#
# Prerequisites:
# - AWS CLI configured with appropriate permissions
# - Slurm version 25.05 or later
# - Root privileges for system configuration
# - Network connectivity to AWS PCS endpoints

set -eo pipefail

# Function to display usage
usage() {
    echo "Usage: $0 --cluster-identifier <cluster-identifier> [--endpoint-url
  <endpoint-url>]"
    echo "    $0 -h|--help"
}

# Function to display help
help() {
    echo "AWS PCS Multi-Cluster Standalone Login Node Configuration Script"
    echo "======"
    echo
    echo "This script configures multi-cluster standalone login node for AWS Parallel
  Computing Service (PCS)"
    echo "by setting up the Slurm authentication and credential kiosk daemon (sackd)."
```

```

# Function to retrieve authentication key
get_auth_key() {
    if [ "$ALTERNATE_SECRET_RETRIEVAL" = "true" ]; then
        echo "Retrieving authentication key from AWS Secrets Manager..." >&2
        local auth_key_arn=$(echo "$CLUSTER_INFO" | jq -r
'.cluster.slurmConfiguration.authKey.secretArn')
        local auth_key_version=$(echo "$CLUSTER_INFO" | jq -r
'.cluster.slurmConfiguration.authKey.secretVersion')

        if [ "$auth_key_arn" = "null" ] || [ "$auth_key_version" = "null" ]; then
            echo "Error: Auth key information not found in cluster configuration" >&2
            exit 1
        fi

        if ! aws secretsmanager get-secret-value --secret-id "$auth_key_arn" --version-
id "$auth_key_version" --query SecretString --output text --region "$REGION" 2>/dev/
null; then
            echo "Error: Failed to retrieve auth key from Secrets Manager" >&2
            exit 1
        fi
    else
        echo "Please enter the base64-encoded Slurm authentication key:" >&2
        echo -n "Base64 of the Slurm secret key: " >&2
        local key
        read -rs key
        echo >&2
        echo "$key"
    fi
}

# Function to get next available SACKD port
get_next_sackd_port() {
    local exclude_file="$1"
    local port=6918
    local used_ports=()

    # Get all currently used SACKD ports into an array
    while IFS= read -r line; do
        used_ports+=("$line")
    done < <(find /etc/sysconfig -name "sackd-pcs-*" ! -path "$exclude_file" \
        -exec grep SACKD_PORT= '{}' ';' 2>/dev/null | \
        sed 's/.*SACKD_PORT=//' | sort -n)

    # Loop through used ports to find first available port

```

```

for used_port in "${used_ports[@]"; do
    if [ "$port" -lt "$used_port" ]; then
        break
    elif [ "$port" -eq "$used_port" ]; then
        ((port++))
    fi
done

echo "$port"
}

# Function to configure cluster
configure_cluster() {
    mkdir -p /etc/slurm
    SLURM_JWKS_FILE="/etc/slurm/slurm-${CLUSTER_NAME}.jwks"
    echo '{"keys":
[{"alg": "HS256", "kty": "oct", "kid": "key-"'${CLUSTER_ID}'"', "k": "'${BASE64_SLURM_KEY}'"}]}'
    | jq -c '.' > "${SLURM_JWKS_FILE}"

    chmod 0600 "$SLURM_JWKS_FILE"
    chown slurm:slurm "$SLURM_JWKS_FILE"

    SLURM_INSTALL_PATH="/opt/aws/pcs/scheduler/slurm-${SLURM_VERSION}"

    SACKD_RUNTIME_DIRECTORY="/run/slurm-${CLUSTER_NAME}"
    mkdir -p "${SACKD_RUNTIME_DIRECTORY}"
    chown slurm:slurm "${SACKD_RUNTIME_DIRECTORY}"

    mkdir -p /etc/sysconfig
    SACKD_SERVICE_NAME="sackd-pcs-${CLUSTER_NAME}"
    SACKD_SERVICE_ENV="/etc/sysconfig/${SACKD_SERVICE_NAME}"
    SACKD_PORT=$(get_next_sackd_port "$SACKD_SERVICE_ENV")
    cat > "${SACKD_SERVICE_ENV}" << EOF
SACKD_OPTIONS='--conf-server=$ENDPOINTS '
SLURM_SACK_JWKS='$SLURM_JWKS_FILE '
RUNTIME_DIRECTORY='$SACKD_RUNTIME_DIRECTORY '
SACKD_PORT=$SACKD_PORT
EOF

    SACKD_SERVICE_PATH="/etc/systemd/system/${SACKD_SERVICE_NAME}.service"

    cat << EOF > "${SACKD_SERVICE_PATH}"
[Unit]
Description=Slurm auth and cred kiosk daemon

```

```

After=network-online.target remote-fs.target
Wants=network-online.target
ConditionPathExists=${SACKD_SERVICE_ENV}

[Service]
Type=notify
EnvironmentFile=${SACKD_SERVICE_ENV}
User=slurm
Group=slurm
RuntimeDirectory=slurm-${CLUSTER_NAME}
RuntimeDirectoryMode=0755
ExecStart=${SLURM_INSTALL_PATH}/sbin/sackd --systemd \${SACKD_OPTIONS}
ExecReload=/bin/kill -HUP \${MAINPID}
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF

    chown root:root "\${SACKD_SERVICE_PATH}"
    chmod 0644 "\${SACKD_SERVICE_PATH}"
    systemctl daemon-reload && systemctl enable "\${SACKD_SERVICE_NAME}"
    systemctl restart "\${SACKD_SERVICE_NAME}"

    ACTIVATE_SCRIPT="activate-pcs-${CLUSTER_NAME}"
    cat > "\${ACTIVATE_SCRIPT}" << EOF
# Activate script for Slurm cluster ${CLUSTER_NAME}

# Add Slurm paths
export PATH="\${SLURM_INSTALL_PATH}/bin:\${PATH}"
export MANPATH="\${SLURM_INSTALL_PATH}/share/man:\${MANPATH}"
export LD_LIBRARY_PATH="\${SLURM_INSTALL_PATH}/lib:\${LD_LIBRARY_PATH}"
ldconfig

# Set Slurm configuration
export SLURM_CONF="/run/slurm-${CLUSTER_NAME}/conf/slurm.conf"
export PCS_CLUSTER_NAME="\${CLUSTER_NAME}"
export PCS_CLUSTER_IDENTIFIER="\${CLUSTER_IDENTIFIER}"
export PCS_CLUSTER_ID="\${CLUSTER_ID}"

echo "Activated PCS cluster environment: ${CLUSTER_NAME}"

```

```

# Deactivate function
function deactivate-pcs- $\{\{\text{CLUSTER\_NAME}\}\}$ () {
    export PATH="\$(echo "\$PATH" | sed -e "s| $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{bin}:||g" -e "s|:
 $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{bin}||g" -e "s|^ $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{bin}\$||")"$ 
    export MANPATH="\$(echo "\$MANPATH" | sed -e "s| $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{share}/\text{man}:||
g" -e "s|: $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{share}/\text{man}||g" -e "s|^ $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{share}/\text{man}\
\$||")"$ 
    export LD_LIBRARY_PATH="\$(echo "\$LD_LIBRARY_PATH" | sed -e "s|
 $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{lib}:||g" -e "s|: $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{lib}||g" -e "s|^
 $\{\{\text{SLURM\_INSTALL\_PATH}\}\}/\text{lib}\$||")"$ 
    unset SLURM_CONF
    unset PCS_CLUSTER_NAME
    unset PCS_CLUSTER_IDENTIFIER
    unset PCS_CLUSTER_ID
    unset -f deactivate-pcs- $\{\{\text{CLUSTER\_NAME}\}\}$ 
    ldconfig
    echo "Deactivated PCS cluster environment:  $\{\{\text{CLUSTER\_NAME}\}\}$ "
}

export -f deactivate-pcs- $\{\{\text{CLUSTER\_NAME}\}\}$ 

EOF
}

# Main function
main() {
    # Parse arguments
    CLUSTER_IDENTIFIER=""
    PCS_ENDPOINT_URL=""

    while [ "$1" != "" ]; do
        case $1 in
            --cluster-identifier)
                shift
                CLUSTER_IDENTIFIER="$1"
                ;;
            --endpoint-url)
                shift
                PCS_ENDPOINT_URL="--endpoint-url $1"
                ;;
            -h|--help)
                help
                exit 0
        esac
    done
}$$$$$$ 
```

```
        ;;
    *)
        echo "Invalid argument: $1" >&2
        usage >&2
        exit 1
        ;;
    esac
    shift
done

# Validate required arguments
if [ -z "$CLUSTER_IDENTIFIER" ]; then
    echo "Error: --cluster-identifier is required" >&2
    usage >&2
    exit 1
fi

# Validate running as root
if [ "$EUID" -ne 0 ]; then
    echo "Error: This script must be run as root" >&2
    exit 1
fi

# Validate required commands are available
for cmd in aws jq curl; do
    if ! command -v "$cmd" &> /dev/null; then
        echo "Error: Required command '$cmd' not found" >&2
        exit 1
    fi
done

# Get the region name from IMDS v2 with error handling (try IPv6 first, fallback to
IPv4)
echo "Retrieving AWS region from instance metadata..."
# Try IPv6 IMDS endpoint first (fd00:ec2::254) with fast timeout (1s connect, 2s
total)
# If IPv6 fails, fallback to IPv4 IMDS endpoint (169.254.169.254)
IMDS_ENDPOINT="http://[fd00:ec2::254]"
if ! TOKEN=$(curl -s -X PUT "${IMDS_ENDPOINT}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --connect-timeout 1 --max-time 2 2>/dev/null); then
    IMDS_ENDPOINT="http://169.254.169.254"
    if ! TOKEN=$(curl -s -X PUT "${IMDS_ENDPOINT}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --max-time 5); then
```

```
        echo "Error: Failed to retrieve IMDS token. Ensure this script is running
on an EC2 instance." >&2
        exit 1
    fi
fi

if ! REGION=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" "${IMDS_ENDPOINT}/
latest/dynamic/instance-identity/document" --max-time 5 | jq -r '.region'); then
    echo "Error: Failed to retrieve AWS region from instance metadata" >&2
    exit 1
fi

echo "Detected AWS region: $REGION"

# Retrieve cluster information from AWS PCS
echo "Retrieving cluster information for: $CLUSTER_IDENTIFIER"
# shellcheck disable=SC2086
if ! CLUSTER_INFO=$(aws pcs get-cluster --region "$REGION" --cluster-identifier
"$CLUSTER_IDENTIFIER" $PCS_ENDPOINT_URL 2>/dev/null); then
    echo "Error: Failed to retrieve cluster information. Check cluster identifier
and AWS permissions." >&2
    exit 1
fi

CLUSTER_ID=$(echo "$CLUSTER_INFO" | jq -r '.cluster.id')
CLUSTER_NAME=$(echo "$CLUSTER_INFO" | jq -r '.cluster.name')
SLURM_VERSION=$(echo "$CLUSTER_INFO" | jq -r '.cluster.scheduler.version')
SLURM_VERSION=${SLURM_VERSION#Slurm_}

# Check if Slurm version is >= 25.05
# shellcheck disable=SC2072
if [[ "$SLURM_VERSION" < "25.05" ]]; then
    echo "Error: This script requires Slurm version 25.05 or later. Found version:
$SLURM_VERSION" >&2
    exit 1
fi

ENDPOINTS=$(echo "$CLUSTER_INFO" | jq -r '.cluster.endpoints[] | select(.type
== "SLURMCTLD") | (if .privateIpAddress != "" then .privateIpAddress else "["
+ .ipv6Address + "]" end) + ":" + .port' | tr '\n' ',' | sed 's/,,$//')

# Get BASE64_SLURM_KEY
BASE64_SLURM_KEY=$(get_auth_key)
```

```
if [ -z "$BASE64_SLURM_KEY" ]; then
    echo "Error: base64 Slurm key cannot be empty" >&2
    exit 1
fi

configure_cluster

# Final configuration summary
echo "======"
echo "Configuration completed successfully!"
echo "======"
echo "Cluster Name: $CLUSTER_NAME"
echo "Cluster ID: $CLUSTER_ID"
echo "Slurm Version: $SLURM_VERSION"
echo "Service Name: $SACKD_SERVICE_NAME"
echo "SACKD Port: $SACKD_PORT"
echo
echo "To activate this cluster environment, run:"
echo "  source ./$ACTIVATE_SCRIPT"
echo
echo "To deactivate this cluster environment, run:"
echo "  deactivate-pcs-`${CLUSTER_NAME}`"
echo
echo "To check service status:"
echo "  systemctl status $SACKD_SERVICE_NAME"
echo
echo "To view service logs:"
echo "  journalctl -u $SACKD_SERVICE_NAME -f"
}

# Exit if being sourced for testing
[[ "${BASH_SOURCE[0]}" != "${0}" ]] && return

# Execute main function
main "$@"
```

使用 AWS PCS 多叢集登入節點組態指令碼

執行指令碼

執行組態指令碼

1. 將[指令碼的內容](#)儲存在名為 的檔案中：

```
pcs-multi-cluster-login-configure.sh
```

2. 使其可執行：

```
chmod +x pcs-multi-cluster-login-configure.sh
```

3. 執行 指令碼：

```
./pcs-multi-cluster-login-configure.sh --cluster-identifier cluster-name
```

叢集互動環境

成功組態後，指令碼會在目前目錄中產生叢集特定的啟用指令碼。指令碼的名稱為 `activate-pcs-cluster-name`。啟用指令碼會設定必要的環境變數和路徑，以與目標叢集互動。

啟用叢集環境

- 使用 `source` 命令來執行啟用指令碼

```
source ./activate-pcs-cluster-name
```

Example

```
# Activate cluster environment for cluster 'my-cluster'  
source ./activate-pcs-my-cluster  
  
# Now you can use Slurm commands  
sinfo  
squeue  
sbatch my-job.sh
```

啟用指令碼的功能

- 設定 `SLURM_CONF` 環境變數以指向叢集的組態。
- 更新 `PATH` 以包含叢集的 Slurm 二進位檔。
- 設定其他必要的 Slurm 環境變數 (`MANPATH`、`LD_LIBRARY_PATH`)。
- 設定 AWS PCS 叢集識別變數。

- 啟用與目標 AWS PCS 叢集的無縫互動。

停用叢集環境

- 執行停用命令。

```
deactivate-pcs-cluster-name
```

Example

```
# After activating a cluster
source ./activate-pcs-my-cluster

# Work with the cluster
sinfo

# Deactivate when done
deactivate-pcs-my-cluster
```

停用命令的功能

- 還原原始PATH環境變數。
- 取消設定叢集特定的 Slurm 環境變數。
- 將 shell 環境傳回至其啟用前狀態。

Note

啟用是工作階段特定的，且必須在您要與叢集互動的 shell 工作階段中取得。

AWS PCS 網路

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

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

主題

- [AWS PCS VPC 和子網路需求和考量事項](#)
- [為您的 AWS PCS 叢集建立 VPC](#)
- [AWS PCS 中的安全群組](#)
- [AWS PCS 中的多個網路介面](#)
- [AWS PCS 中 EC2 執行個體的置放群組](#)
- [搭配 AWS PCS 使用 Elastic Fabric Adapter \(EFA\)](#)

AWS PCS VPC 和子網路需求和考量事項

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

VPC 要求和注意事項

當您建立叢集時，您指定的 VPC 必須符合下列要求和注意事項：

- VPC 必須有足夠數量的 IP 地址，可供您要建立的叢集、任何節點和其他叢集資源使用。如需詳細資訊，請參閱《Amazon [VPCs 使用者指南](#)》中的 [VPC 和子網路的 IP 定址](#)。
- 如果您的叢集使用 IPv6：
 - 建立 IPv6 CIDR 區塊與 VPC 的關聯。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [建立 VPC](#)。

Important

雖然您可以使用 IPv4 和 IPv6 設定 VPC，但您只能為叢集選擇 1 個網路類型。

- 為您的子網路啟用自動指派 IPv6 地址。
- 如需詳細資訊，請參閱：
 - [上的 IPv6 AWS](#)
 - [了解 AWS 上的 IPv6 定址，並設計可擴展的定址計畫](#)
- VPC 必須具有 DNS 主機名稱和 DNS 解析支援。否則，節點無法註冊客戶叢集。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [VPC 的 DNS 屬性](#)。
- VPC 可能需要使用的 VPC AWS PrivateLink 端點才能聯絡 AWS PCS API。如需詳細資訊，請參閱《Amazon [VPC 使用者指南](#)》中的使用 [將您的 VPC 連接到 服務 AWS PrivateLink](#)。

Important

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

子網需求和注意事項

當您建立 Slurm 叢集時，AWS PCS 會在您指定的子網路中建立 [彈性網路界面 \(ENI\)](#)。此網路界面可啟用排程器控制器與客戶 VPC 之間的通訊。網路界面也可讓 Slurm 與您帳戶中部署的元件進行通訊。您只能在建立時指定叢集的子網路。

叢集的子網路要求

您在建立叢集時指定的 [子網路](#) 必須符合下列要求：

- 子網路必須至少有 1 個 IP 地址供 AWS PCS 使用。
- 如果您的叢集使用 IPv6，叢集中的所有子網路都必須使用 IPv6。

Important

如果子網路只設定為使用 IPv6，則使用 AWS PCS 範例 AMIs 和多個網路介面設定的運算節點群組目前無法運作。請改用雙堆疊子網路 (IPv4 和 IPv6) IPv4-only 的子網路。如需詳細資訊，請參閱 [搭配 AWS PCS 使用範例 Amazon Machine Image AMIs](#)。

- 子網路不能位於 AWS Outposts、AWS Wavelength 或 AWS Local Zone。
- 子網路可以是公有或私有。如果可能，我們建議您指定私有子網路。公有子網路是具有包含 [網際網路閘道](#) 路由之路由表的子網路；私有子網路是具有不包含網際網路閘道路由之路由表的子網路。

節點的子網路要求

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

您部署節點和叢集資源到 的任何子網路必須符合下列要求：

- 您必須確保子網路有足夠的可用 IP 地址來部署所有節點和叢集資源。
- 如果您的叢集使用 IPv4，且您打算將節點部署到公有子網路，則該子網路必須自動指派 IPv4 公有地址。

Note

公有子網路中的執行個體必須使用具有允許來自公有 IP 地址流量之傳入規則的安全群組。除非您有特定的來源地址限制，否則這表示 IPv4 來源地址為 0.0.0.0/0 或 IPv6 來源地址為 ::/0。

- 如果您部署節點的子網路是私有子網路，且其路由表不包含網路位址轉譯 ([NAT](#)) 裝置 (IPv4) 的路由，請使用 AWS PrivateLink 將 VPC 端點新增至客戶 VPC。節點聯絡的所有 AWS 服務都需要 VPC 端點。唯一的必要端點是 AWS PCS 允許節點呼叫 RegisterComputeNodeGroupInstance API 動作。如需詳細資訊，請參閱 AWS PCS API 參考中的 [RegisterComputeNodeGroupInstance](#)。
- 公有或私有子網路狀態不會影響 AWS PCS；必要的端點必須能夠連線。

為您的 AWS PCS 叢集建立 VPC

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

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

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

先決條件

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

建立 Amazon VPC

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

- US East (N. Virginia) (美國東部 (維吉尼亞北部)) (us-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- US East (Ohio) (美國東部 (俄亥俄)) (us-east-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- US West (Oregon) (美國西部 (奧勒岡)) (us-west-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-west-2#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asia Pacific (Mumbai) (亞太區域 (孟買)) (ap-south-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-south-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asia Pacific (Singapore) (亞太區域 (新加坡)) (ap-southeast-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asia Pacific (Sydney) (亞太區域 (雪梨)) (ap-southeast-2)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-2#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asia Pacific (Tokyo) (亞太區域 (東京)) (ap-northeast-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-northeast-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 歐洲 (法蘭克福) (eu-central-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-central-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 歐洲 (愛爾蘭) (eu-west-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 歐洲 (倫敦) (eu-west-2)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-west-2#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 歐洲 (巴黎) (eu-west-3)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-west-3#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 歐洲 (米蘭) (eu-south-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-south-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 歐洲 (斯德哥爾摩) (eu-north-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-north-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- AWS GovCloud (美國東部) (us-gov-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-gov-east-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- AWS GovCloud (美國西部) (us-gov-west-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-gov-west-1#/stacks/create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 僅範本

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

為 PCS 建立 Amazon VPC


1. 在 [CloudFormation 主控台](#) 中開啟 範本。

Note

這些項目會預先填入範本中，讓您可以直接將它們保留為預設值。

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


- a. 在 VPC 下，接著輸入 CidrBlock 10.3.0.0/16
- b. 在子網路 A 下：
 - i. 然後 CidrPublicSubnetA，輸入 10.3.0.0/20
 - ii. 然後 CidrPrivateSubnetA，輸入 10.3.128.0/20
- c. 在子網路 B 下：
 - i. 然後 CidrPublicSubnetB，輸入 10.3.16.0/20
 - ii. 然後 CidrPrivateSubnetA，輸入 10.3.144.0/20
- d. 在子網路 C 下：
 - i. 針對 ProvisionSubnetsC，選取 True。

 Note

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

- ii. 然後 CidrPublicSubnetB，輸入 10.3.32.0/20
 - iii. 然後 CidrPrivateSubnetA，輸入 10.3.160.0/20
4. 在功能下，勾選我確認 AWS CloudFormation 可能會建立 IAM 資源的核取方塊。

監控 CloudFormation 堆疊的狀態。到達 時 `CREATE_COMPLETE`，VPC 資源已準備好供您使用。

 Note

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

AWS PCS 中的安全群組

Amazon EC2 中的安全群組充當虛擬防火牆，以控制執行個體的傳入和傳出流量。使用 AWS PCS 運算節點群組的啟動範本來新增或移除其執行個體的安全群組。如果您的啟動範本不包含任何網路介面，請使用 `SecurityGroupIds` 提供安全群組的清單。如果您的啟動範本定義網路介面，您必須使用

Groups 參數將安全群組指派給每個網路介面。如需啟動範本的詳細資訊，請參閱 [搭配 AWS PCS 使用 Amazon EC2 啟動範本](#)。

Note

對啟動範本中安全群組組態的變更只會影響運算節點群組更新後啟動的新執行個體。

安全群組需求和考量事項

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

達成此目標的直接方式是建立寬鬆的自我參考安全群組，允許群組所有成員之間所有連接埠上的 TCP/IP 流量。您可以將此連接到叢集和節點群組 EC2 執行個體。

允許的安全群組組態範例

IPv4

規則類型	通訊協定	連接埠	來源	目標
傳入	全部	全部	Self	
傳出	全部	全部		0.0.0.0/0
傳出	全部	全部		Self

IPv6

規則類型	通訊協定	連接埠	來源	目標
傳入	全部	全部	Self	
傳出	全部	全部		::/0
傳出	全部	全部		Self

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

限制性安全群組組態範例

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

- 6817 – 啟用slurmctld從 EC2 執行個體到 的傳入連線
- 6818 – 啟用從 slurmctld 到 EC2 執行個體slurmd上執行的傳出連線

連接至運算節點的安全群組必須允許下列連接埠：

- 6817 – 啟用slurmctld從 EC2 執行個體到 的傳出連線。
- 6818 – 在節點群組執行個體slurmd上啟用往返 slurmdslurmctld的傳入和傳出連線
- 60001–63000 – 要支援的節點群組執行個體之間的傳入和傳出連線 srun
- 節點群組執行個體之間的 EFA 流量。如需詳細資訊，請參閱《Linux 執行個體使用者指南》中的[準備啟用 EFA 的安全群組](#)
- 工作負載所需的任何其他節點間流量

AWS PCS 中的多個網路介面

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

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

- 每個網路界面的子網路必須與您在設定將使用啟動範本的 AWS PCS 運算節點群組時所選擇的子網路相同。
- 主要網路裝置，例如 SSH 和 HTTPS 流量等例行網路通訊，是透過將 設定為 DeviceIndex 來建立 0。其他網路介面具有 DeviceIndex 的 1。只能有一個主要網路介面，所有其他介面都是次要的。
- 所有網路介面都必須具有唯一的 NetworkCardIndex。建議的做法是按啟動範本中定義的順序對它們進行編號。

- 每個網路界面的安全群組都是使用 `設定Groups`。在此範例中，傳入 SSH 安全群組 (`sg-SshSecurityGroupId`) 會新增至主要網路介面，以及啟用叢集內通訊 () 的安全群組 `sg-ClusterSecurityGroupId`。最後，允許對外連線至網際網路 (`sg-InternetOutboundSecurityGroupId`) 的安全群組會同時新增至主要和次要介面。

```
{
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "NetworkCardIndex": 0,
      "SubnetId": "subnet-SubnetId",
      "Groups": [
        "sg-SshSecurityGroupId",
        "sg-ClusterSecurityGroupId",
        "sg-InternetOutboundSecurityGroupId"
      ]
    },
    {
      "DeviceIndex": 1,
      "NetworkCardIndex": 1,
      "SubnetId": "subnet-SubnetId",
      "Groups": ["sg-InternetOutboundSecurityGroupId"]
    }
  ]
}
```

AWS PCS 中 EC2 執行個體的置放群組

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

置放群組類型

- 叢集：將執行個體封裝在可用區域中，以最佳化低延遲通訊。
- 分割區 – 將執行個體分散到邏輯分割區，以協助最大化彈性。
- 分散 – 嚴格強制少數執行個體在不同的硬體上啟動，這也有助於恢復能力。

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

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

建立使用 EFA 的叢集置放群組

1. 使用運算節點群組的類型叢集建立置放群組。

- 使用下列 AWS CLI 命令：

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

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

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

2. 在 AWS PCS 運算節點群組的 EC2 啟動範本中包含置放群組。

搭配 AWS PCS 使用 Elastic Fabric Adapter (EFA)

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

Note

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

內容

- [識別啟用 EFA 的 EC2 執行個體](#)
- [建立安全群組以支援 EFA 通訊](#)
- [\(選用\) 建立置放群組](#)
- [建立或更新 EC2 啟動範本](#)

- [建立或更新 EFA 的運算節點群組](#)
- [\(選用\) 測試 EFA](#)
- [\(選用\) 使用 CloudFormation 範本建立啟用 EFA 的啟動範本](#)

識別啟用 EFA 的 EC2 執行個體

若要使用 EFA，AWS PCS 運算群組允許的所有執行個體類型都必須支援 EFA，且必須具有相同數量 vCPUs（和 GPUs，如適用）。如需已啟用 EFA 的執行個體清單，請參閱《[Amazon Elastic Compute Cloud 使用者指南](#)》中的適用於 Amazon EC2 上 HPC 和 ML 工作負載的 [Elastic Fabric Adapter](#)。您也可以使用 AWS CLI 來檢視支援 EFA 的執行個體類型清單。將 *region-code* 取代為您使用 AWS PCS AWS 區域的，例如 us-east-1。

```
aws ec2 describe-instance-types \
  --region region-code \
  --filters Name=network-info.efa-supported,Values=true \
  --query "InstanceTypes[*].[InstanceType]" \
  --output text | sort
```

Note

確定有多少網路介面可用 – 有些 EC2 執行個體具有多個網路卡。這可讓它們擁有多個 EFAs。如需詳細資訊，請參閱[AWS PCS 中的多個網路介面](#)。

建立安全群組以支援 EFA 通訊

AWS CLI

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

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

```
aws ec2 create-security-group \
```

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

使用下列命令來連接傳入和傳出安全群組規則。進行下列取代：

- *efa-secgroup-id* – 提供您剛建立的 EFA 安全群組 ID。

```
aws ec2 authorize-security-group-ingress \  
  --group-id efa-secgroup-id \  
  --protocol -1 \  
  --source-group efa-secgroup-id  
  
aws ec2 authorize-security-group-egress \  
  --group-id efa-secgroup-id \  
  --protocol -1 \  
  --source-group efa-secgroup-id
```

CloudFormation template

您可以使用 CloudFormation 範本來建立支援 EFA 的安全群組。從下列 URL 下載範本，然後將其上傳至 [AWS CloudFormation 主控台](#)。

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

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

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

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

(選用) 建立置放群組

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

建立或更新 EC2 啟動範本

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

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

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

```
{
  "Placement": {
    "GroupId": "pg-PlacementGroupId1"
  },
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "InterfaceType": "efa",
      "NetworkCardIndex": 0,
      "SubnetId": "subnet-SubnetID1",
      "Groups": [
        "sg-SecurityGroupId1",
        "sg-EfaSecGroupId"
      ]
    },
    {
      "DeviceIndex": 1,
      "InterfaceType": "efa",
      "NetworkCardIndex": 1,
      "SubnetId": "subnet-SubnetID1"
      "Groups": ["sg-EfaSecGroupId"]
    }
  ]
}
```

```
]
}
```

建立或更新 EFA 的運算節點群組

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

(選用) 測試 EFA

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

若要開始，您需要在運算節點群組中執行兩個執行個體。如果您的運算節點群組使用靜態容量，則應該已有可用的執行個體。對於使用動態容量的運算節點群組，您可以使用 `salloc` 命令啟動兩個節點。以下是來自叢集的範例，該叢集具有名為 `dynamic` 的動態節點群組，該群組與名為 `hpc7g` 的佇列相關聯。

```
% salloc --nodes 2 -p all
salloc: Granted job allocation 6
salloc: Waiting for resource configuration
... a few minutes pass ...
salloc: Nodes hpc7g-[1-2] are ready for job
```

使用 `scontrol` 找出兩個配置節點的 IP 地址。在下列範例中，地址 `10.3.140.69` 適用於 `hpc7g-1`，而 `10.3.132.211` 適用於 `hpc7g-2`。

```
% scontrol show nodes hpc7g-[1-2]
NodeName=hpc7g-1 Arch=aarch64 CoresPerSocket=1
  CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
  AvailableFeatures=hpc7g
  ActiveFeatures=hpc7g
  Gres=(null)
  NodeAddr=10.3.140.69 NodeHostName=ip-10-3-140-69 Version=25.05.5
  OS=Linux 5.10.218-208.862.amzn2.aarch64 #1 SMP Tue Jun 4 16:52:10 UTC 2024
  RealMemory=124518 AllocMem=0 FreeMem=110763 Sockets=64 Boards=1
  State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
  Partitions=efa
  BootTime=2024-07-02T19:00:09 SlurmdStartTime=2024-07-08T19:33:25
  LastBusyTime=2024-07-08T19:33:25 ResumeAfterTime=None
```

```

CfgTRES=cpu=64,mem=124518M,billing=64
AllocTRES=
CapWatts=n/a
CurrentWatts=0 AveWatts=0
ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
Reason=Maintain Minimum Number Of Instances [root@2024-07-02T18:59:00]
InstanceId=i-04927897a9ce3c143 InstanceType=hpc7g.16xlarge

NodeName=hpc7g-2 Arch=aarch64 CoresPerSocket=1
CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
AvailableFeatures=hpc7g
ActiveFeatures=hpc7g
Gres=(null)
NodeAddr=10.3.132.211 NodeHostName=ip-10-3-132-211 Version=25.05.5
OS=Linux 5.10.218-208.862.amzn2.aarch64 #1 SMP Tue Jun 4 16:52:10 UTC 2024
RealMemory=124518 AllocMem=0 FreeMem=110759 Sockets=64 Boards=1
State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=efa
BootTime=2024-07-02T19:00:09 SlurmdStartTime=2024-07-08T19:33:25
LastBusyTime=2024-07-08T19:33:25 ResumeAfterTime=None
CfgTRES=cpu=64,mem=124518M,billing=64
AllocTRES=
CapWatts=n/a
CurrentWatts=0 AveWatts=0
ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
Reason=Maintain Minimum Number Of Instances [root@2024-07-02T18:59:00]
InstanceId=i-0a2c82623cb1393a7 InstanceType=hpc7g.16xlarge

```

使用 SSH (或 SSMhpc7g-1) 連線到其中一個節點 (在此範例中為)。請注意，這是內部 IP 地址，因此如果您使用 SSH，您可能需要從其中一個登入節點進行連線。另請注意，執行個體需要透過運算節點群組啟動範本使用 SSH 金鑰進行設定。

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

現在，fi_pingpong以伺服器模式啟動。

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

連線至第二個執行個體 (hpc7g-2)。

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

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

```
% /opt/amazon/efa/bin/fi_pingpong -p efa 10.3.140.69

bytes   #sent   #ack    total      time      MB/sec    usec/xfer  Mxfers/sec
64      10      =10     1.2k       0.00s     3.08     20.75     0.05
256     10      =10     5k         0.00s     21.24    12.05     0.08
1k      10      =10     20k        0.00s     82.91    12.35     0.08
4k      10      =10     80k        0.00s     311.48   13.15     0.08
[error] util/pingpong.c:1876: fi_close (-22) fid 0
```

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

由於設定 EFA 有數個相依性，因此已提供 CloudFormation 範本，供您用來設定運算節點群組。它支援最多四個網路卡的執行個體。若要進一步了解具有多個網路卡的執行個體，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的彈性[網路介面](#)。

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

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

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

- 在提供堆疊名稱下
 - 在堆疊名稱下，輸入描述性名稱。我們建議您合併為 AWS PCS 運算節點群組選擇的名稱，例如 ***NODEGROUPNAME-efa-lt***。
- 在參數下
 - 在 NumberOfNetworkCards 下，選擇節點群組中執行個體中的網路卡數量。
 - 在 VpcId 下，選擇 AWS PCS 叢集部署所在的 VPC。
 - 在 NodeGroupSubnetId 下，選擇叢集 VPC 中要啟動啟用 EFA 執行個體的子網路。
 - 在 PlacementGroupName 下，將欄位保留空白，以為節點群組建立新的叢集置放群組。如果您有想要使用的現有置放群組，請在此處輸入其名稱。
 - 在 ClusterSecurityGroupId 下，選擇您用來允許存取叢集中其他執行個體和 AWS PCS API 的安全群組。許多客戶從其叢集 VPC 選擇預設安全群組。

- 在 SshSecurityGroupId 下，提供您用來允許傳入 SSH 存取叢集中節點的安全群組 ID。
- 針對 SshKeyName，選取 SSH 金鑰對以存取叢集中的節點。
- 針對 LaunchTemplateName，輸入啟動範本的描述性名稱，例如 *NODEGROUPNAME-efa-1t*。名稱對於您將使用 AWS PCS AWS 區域的 AWS 帳戶 必須是唯一的。
- 在功能下
 - 勾選核取方塊以確認 AWS CloudFormation 可能會建立 IAM 資源。

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

搭配 AWS PCS 使用網路檔案系統

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

本主題涵蓋搭配 AWS PCS 使用網路檔案系統的考量和範例。

使用網路檔案系統的考量事項

各種檔案系統的實作詳細資訊不同，但有一些常見的考量。

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

網路掛載範例

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

Amazon EFS

檔案系統設定

建立 Amazon EFS 檔案系統。請確定它在您要啟動 PCS 運算節點群組執行個體的每個可用區域中都有掛載目標。同時確保每個掛載目標都與允許從 PCS 運算節點群組執行個體傳入和傳出存取的安全群

組相關聯。如需詳細資訊，請參閱《Amazon Elastic File System 使用者指南》中的[掛載目標和安全群組](#)。

啟動範本

將安全群組（從您的檔案系統設定）新增至您將用於運算節點群組的啟動範本。

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

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

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

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

packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /mount-point-directory
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults

--==MYBOUNDARY==--
```

Amazon FSx for Lustre

檔案系統設定

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

FSx

啟動範本

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

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

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

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

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

--===MYBOUNDARY===--
```

Amazon FSx for NetApp ONTAP

檔案系統設定

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

啟動範本

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

- *mount-point-directory* – 您想要掛載 FSx for ONTAP 磁碟區的執行個體路徑

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

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

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

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

--==MYBOUNDARY==--
```

Amazon FSx for OpenZFS

檔案系統設定

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

啟動範本

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

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

```
MIME-Version: 1.0
```

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

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

runcmd:
- mkdir -p /mount-point-directory
- mount -t nfs -o noatime,nfsvers=4.2,sync,rsize=1048576,wsiz=1048576 filesystem-id.fsx.region-code.amazonaws.com:/fsx/ /mount-point-directory

--===MYBOUNDARY===--
```

Amazon File Cache

檔案系統設定

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

啟動範本

將安全群組（從您的檔案系統設定）新增至您將用於運算節點群組的啟動範本。

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

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

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

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

runcmd:
- amazon-linux-extras install -y lustre=2.12
- mkdir -p /mount-point-directory
```

```
- mount -t lustre -o relatime,flock cache-dns-name@tcp:/mount-name /mount-point-  
directory  
  
--==MYBOUNDARY==--
```

AWS PCS 的 Amazon Machine Image AMIs)

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

主題

- [搭配 AWS PCS 使用範例 Amazon Machine Image AMIs\)](#)
- [AWS PCS 的自訂 Amazon Machine Image AMIs\)](#)
- [建置 AWS PCS 自訂 AMIs 的軟體安裝程式](#)
- [AWS PCS 範例 AMIs 版本備註](#)

搭配 AWS PCS 使用範例 Amazon Machine Image AMIs)

AWS 提供[範例 AMIs](#)，可用來做為使用 AWS PCS 的起點。

Important

範例 AMIs 僅供示範之用，不建議用於生產工作負載。

Important

如果子網路只設定為使用 IPv6，則使用 AWS PCS 範例 AMIs 和多個網路介面設定的運算節點群組目前無法運作。請改用雙堆疊子網路 (IPv4 和 IPv6) IPv4-only 的子網路。

尋找目前的 AWS PCS 範例 AMIs

AWS 管理主控台

AWS PCS 範例 AMIs 具有下列命名慣例：

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

接受的值

- `####` – amzn2
- `##` – x86_64 或 arm64
- `###` – slurm
- `scheduler-major-version` – 25.05

尋找 AWS PCS 範例 AMIs

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

範例

- Arm64 執行個體上 Slurm 25.05 的範例 AMI

```
aws-pcs-sample_ami-amzn2-arm64-slurm-25.05
```

- 在 x86 執行個體上 Slurm 25.05 的範例 AMI

```
aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05
```

Note

如果有多個 AMIs，請使用具有最新時間戳記的 AMI。

5. 當您建立或更新運算節點群組時，請使用 AMI ID。

AWS CLI

您可以使用以下命令找到最新的 AWS PCS 範例 AMI。將 `region-code` 取代為您使用 AWS PCS AWS 區域的，例如 `us-east-1`。

- `x86_64`

```
aws ec2 describe-images --region region-code --owners amazon \  
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05*' \  
         'Name=state,Values=available' \  
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

- Arm64

```
aws ec2 describe-images --region region-code --owners amazon \  
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-arm64-slurm-25.05*' \  
         'Name=state,Values=available' \  
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

當您建立或更新運算節點群組時，請使用 AMI ID。

進一步了解 AWS PCS 範例 AMIs

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

建置與 AWS PCS 相容的自有 AMIs

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

AWS PCS 的自訂 Amazon Machine Image AMIs)

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

Note

如果您想要在不建置自訂 AMI 的情況下嘗試 AWS PCS，您可以使用提供的範例 AMI AWS。如需詳細資訊，請參閱 [搭配 AWS PCS 使用範例 Amazon Machine Image AMIs](#)。

Important

AWS PCS 目前需要支援 IPv4 的核心進行本機節點通訊，即使您在 IPv6-only 的網路中使用 AWS PCS 也是如此。

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

主題

- [步驟 1 – 啟動暫時執行個體](#)
- [步驟 2 – 安裝 AWS PCS 代理程式](#)
- [步驟 3 – 安裝 Slurm](#)
- [步驟 4 – \(選用\) 安裝其他驅動程式、程式庫和應用程式軟體](#)
- [步驟 5 – 建立與 AWS PCS 相容的 AMI](#)
- [步驟 6 – 搭配 AWS PCS 運算節點群組使用自訂 AMI](#)
- [步驟 7 – 終止暫時執行個體](#)

步驟 1 – 啟動暫時執行個體

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

啟動暫時執行個體

1. 開啟 [Amazon EC2 主控台](#)。
2. 在導覽窗格中，選擇執行個體，然後選擇啟動執行個體以開啟新的啟動執行個體精靈。
3. (選用) 在名稱和標籤區段中，提供執行個體的名稱，例如 PCS-AMI-instance。該名稱將指派作為執行個體的資源標籤 (Name=PCS-AMI-instance)。
4. 在 Application and OS Images (應用程式和作業系統映像) 區段中，為其中一個 [支援的作業系統](#) 選取 AMI。
5. 在 Instance type (執行個體類型) 區段中，選取 [支援的執行個體類型](#)。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在網路設定區段中：

- 針對防火牆（安全群組），選擇選取現有的安全群組，然後選取允許傳入 SSH 存取執行個體的安全群組。
8. 在儲存區段中，根據需求設定磁碟區。請務必設定足夠的空間來安裝您自己的應用程式和程式庫。
 9. 在 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

步驟 2 – 安裝 AWS PCS 代理程式

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

安裝 AWS PCS 代理程式

1. 連接至您啟動的執行個體。如需詳細資訊，請參閱連線至 Linux 執行個體。
2. （選用）為了確保您的所有軟體套件都是最新的，請在執行個體上執行快速軟體更新。此程序可能需要幾分鐘的時間。
 - Amazon Linux 2、Amazon Linux 2023、RHEL 9、RHEL 8、Rocky Linux 9 和 Rocky Linux 8

```
sudo yum update -y
```


- Ubuntu 22.04 和 Ubuntu 24.04

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

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

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz -o aws-pcs-agent-v1.3.2-1.tar.gz
```

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

 Note

這可能會在 AWS PCS 代理程式軟體的未來版本中變更。

5. (選用) 驗證 AWS PCS 軟體 tarball 的真實性和完整性。我們建議您執行這項操作來確認軟體發布者的身分，並檢查檔案自發行以來並未遭到變更或損毀。
 - a. 下載 AWS PCS 的公有 GPG 金鑰，並將其匯入至 keyring。使用啟動暫時執行個體 AWS 區域的取代##。命令應傳回金鑰值。記錄索引鍵值；您可以在下一個步驟中使用它。


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

- b. 執行下列命令來驗證 GPG 金鑰的指紋。

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

命令應傳回與下列項目相同的指紋：

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

 Important

如果指紋不相符，請勿執行 AWS PCS 代理程式安裝指令碼。聯絡 [AWS 支援](#)。

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

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz.sig && \  
  gpg --verify ./aws-pcs-agent-v1.3.2-1.tar.gz.sig
```

輸出格式應類似以下內容：

```
gpg: assuming signed data in './aws-pcs-agent-v1.3.2-1.tar.gz'  
gpg: Signature made Thu 06 Nov 2025 11:10:36 AM CET using RSA key ID ECC0AE5C  
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"
```

```
gpg: WARNING: This key is not certified with a trusted signature!  
gpg:          There is no indication that the signature belongs to the owner.  
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A 239A 7EEF 030E DDF5 C21C  
Subkey fingerprint: B7E1 8788 3517 6A74 C3D5 EAF5 6088 136D ECC0 AE5C
```

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

⚠ Important

如果指紋不相符，請勿執行 AWS PCS 軟體安裝指令碼。聯絡 [AWS 支援](#)。

6. 從壓縮 `.tar.gz` 檔案擷取檔案並導覽至擷取的目錄。

```
tar -xf aws-pcs-agent-v1.3.2-1.tar.gz && \  
cd aws-pcs-agent
```

7. 安裝 AWS PCS 軟體。

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

8. 檢查 AWS PCS 軟體版本檔案以確認安裝成功。

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

輸出格式應類似以下內容：

```
AGENT_INSTALL_DATE='Fri Dec 13 12:28:43 UTC 2024'  
AGENT_VERSION='1.3.2'  
AGENT_RELEASE='1'
```

步驟 3 – 安裝 Slurm

安裝與 AWS PCS 相容的 Slurm 版本。如需詳細資訊，請參閱 [AWS PCS 中的 Slurm 版本](#)。

Note

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

安裝 Slurm

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

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz \  
-o aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz
```

您也可以在上述命令latest中，將版本編號取代為，以取得最新版本（例如：aws-pcs-slurm-25.05-installer-latest.tar.gz）。如需具有檢查總和之可用版本的完整清單，請參閱 [AWS PCS 中的 Slurm 版本](#)。

Note

這可能會在 Slurm 安裝程式軟體的未來版本中變更。

3. （選用）驗證 Slurm 安裝程式 tarball 的真實性和完整性。我們建議您執行這項操作來確認軟體發布者的身分，並檢查檔案自發行以來並未遭到變更或損毀。
 - a. 下載 AWS PCS 的公有 GPG 金鑰，並將其匯入至 keyring。使用啟動暫時執行個體 AWS 區域的 取代##。命令應傳回金鑰值。記錄索引鍵值；您可以在下一個步驟中使用它。

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

- b. 執行下列命令來驗證 GPG 金鑰的指紋。

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

命令應傳回與下列項目相同的指紋：

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

⚠ Important

如果指紋不相符，請勿執行 Slurm 安裝指令碼。聯絡 [AWS 支援](#)。

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

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz.sig && \  
gpg --verify ./aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz.sig
```

輸出格式應類似以下內容：

```
gpg: assuming signed data in './aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz'  
gpg: Signature made Fri 14 Nov 2025 11:35:15 AM UTC using RSA key ID ECC0AE5C  
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg:          There is no indication that the signature belongs to the owner.  
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A 239A 7EEF 030E DDF5 C21C  
Subkey fingerprint: B7E1 8788 3517 6A74 C3D5 EAF5 6088 136D ECC0 AE5C
```

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

⚠ Important

如果指紋不相符，請勿執行 Slurm 安裝指令碼。聯絡 [AWS 支援](#)。

4. 從壓縮的 .tar.gz 檔案中解壓縮檔案，然後導覽至解壓縮的目錄。

```
tar -xf aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz && \  
cd aws-pcs-slurm-25.05-installer
```

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

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

6. 檢查排程器版本檔案以確認安裝。

```
cat /opt/aws/pcs/scheduler/slurm-25.05/version
```

輸出格式應類似以下內容：

```
SLURM_INSTALL_DATE='Fri Nov 14 15:15:37 UTC 2025'  
SLURM_VERSION='25.05.5'  
PCS_SLURM_RELEASE='1'
```

步驟 4 – (選用) 安裝其他驅動程式、程式庫和應用程式軟體

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

範例

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

步驟 5 – 建立與 AWS PCS 相容的 AMI

安裝必要的軟體元件之後，您可以建立 AMI，以便重複使用它在 AWS PCS 運算節點群組中啟動執行個體。

Important

AWS PCS 目前需要支援 IPv4 的核心進行本機節點通訊，即使您在 IPv6-only 的網路中使用 AWS PCS 也是如此。

從暫時執行個體建立 AMI

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

步驟 6 – 搭配 AWS PCS 運算節點群組使用自訂 AMI

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

Important

AWS PCS 目前需要支援 IPv4 的核心進行本機節點通訊，即使您在 IPv6-only 的網路中使用 AWS PCS 也是如此。

New compute node group

使用自訂 AMI

1. 開啟 [AWS PCS 主控台](#)。
2. 在導覽窗格中，選擇叢集。
3. 選擇您將使用自訂 AMI 的叢集，然後選取運算節點群組。
4. 建立新的運算節點群組。如需詳細資訊，請參閱 [在 AWS PCS 中建立運算節點群組](#)。在 AMI ID 下，搜尋您要使用的自訂 AMI 的名稱或 ID。完成設定運算節點群組，然後選擇建立運算節點群組。
5. （選用）確認 AMI 支援執行個體啟動。在運算節點群組中啟動執行個體。您可以透過將運算節點群組設定為具有單一靜態執行個體來執行此操作，也可以將任務提交至使用運算節點群組的佇列。
 - a. 檢查 Amazon EC2 主控台，直到執行個體出現標記新的運算節點群組 ID。如需詳細資訊，請參閱 [尋找 AWS PCS 中的運算節點群組執行個體](#)。
 - b. 當您看到執行個體啟動並完成其引導程序時，請確認它正在使用預期的 AMI。若要這樣做，請選取執行個體，然後在詳細資訊下檢查 AMI ID。它應該符合您在運算節點群組設定中設定的 AMI。
 - c. （選用）將運算節點群組擴展組態更新為您偏好的值。

Existing compute node group

使用自訂 AMI

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

- a. 檢查 Amazon EC2 主控台，直到執行個體出現標記新的運算節點群組 ID。如需詳細資訊，請參閱 [尋找 AWS PCS 中的運算節點群組執行個體](#)。
- b. 當您看到執行個體啟動並完成其引導程序時，請確認它正在使用預期的 AMI。若要這樣做，請選取執行個體，然後在詳細資訊下檢查 AMI ID。它應該符合您在運算節點群組設定中設定的 AMI。
- c. (選用) 將運算節點群組擴展組態更新為您偏好的值。

步驟 7 – 終止暫時執行個體

在您確認 AMI 與 AWS PCS 如預期般運作後，您可以終止暫時執行個體來停止產生費用。

終止暫時執行個體

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

建置 AWS PCS 自訂 AMIs 的軟體安裝程式

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

內容

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

AWS PCS 代理程式軟體安裝程式

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

如需 AWS PCS 代理程式軟體的詳細資訊，請參閱 [AWS PCS 代理程式版本](#)。

Slurm 安裝程式

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

AWS 提供的軟體會安裝下列項目：

- 請求的主要和維護版本（目前為 25.05.x 版）上的 [Slurm - 授權 GPL 2](#)
 - Slurm 建置時將 `--sysconfdir` 設為 `/etc/slurm`
 - Slurm 是使用選項 `--enable-pam` 和建置的 `--without-munge`
 - Slurm 是使用選項建置 `--sharedstatedir=/run/slurm/`
 - Slurm 使用 PMIX 和 JWT 支援建置
 - Slurm 安裝在 `/opt/aws/pcs/schedulers/slurm-25.05`
- [OpenPMIX](#) (4.2.6 版) – [授權](#)
 - OpenPMIX 已安裝為的子目錄 `/opt/aws/pcs/scheduler/`
- [libjwt](#) (1.17.0 版) – [授權 MPL-2.0](#)
 - libjwt 已安裝為的子目錄 `/opt/aws/pcs/scheduler/`

AWS 提供的軟體會變更系統組態，如下所示：

- 組建建立的 Slurm systemd 檔案會以 `/etc/systemd/system/` 檔案名稱複製到 `slurmd-25.05.service`。
- 如果不存在，則會使用 UID/GID 建立 Slurm 使用者和群組 (`slurm:slurm`)401。
- 資料夾 `/etc/aws/pcs/scheduler/slurm-25.05/plugstack.conf.d/` 已建立，以存放您的 [使用 SPANK 外掛程式擴展 AWS PCS 上的 Slurm 功能](#) 組態。
- 在 Amazon Linux 2 和 Rocky Linux 9 上，安裝會新增 EPEL 儲存庫，以安裝建置 Slurm 或其相依性所需的軟體。

- 在 RHEL9 上，安裝將啟用 `fedoraproject codeready-builder-for-rhel-9-rhui-rpms` 和 `epel-release-latest-9` 從安裝必要的軟體，以建置 Slurm 或其相依性。

支援的作業系統

請參閱 [AWS PCS 中支援的作業系統](#)。

Note

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

支援的執行個體類型

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

支援的 Slurm 版本

請參閱 [AWS PCS 中的 Slurm 版本](#)。

使用檢查總和驗證安裝程式

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

驗證 tarball

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

- SHA256

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

命令應以下列格式傳回檢查總和值。

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

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

Important

如果檢查總和不相符，請勿執行安裝指令碼。請聯絡 [支援](#)。

例如，下列命令會產生 Slurm 25.05.5-1 tarball 的 SHA256 檢查總和。

```
$ sha256sum aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz
```

輸出範例：

```
3b0f93bce441d4f4f6935175f2c1e81cd961cb923adb416fa6689f5592047a7d aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz
```

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

AWS PCS 代理程式

Installer (安裝程式)	下載 URL	SHA256 檢查總和
AWS PCS 代理程式 1.3.2-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz</pre>	<pre>06b32a952a1c849e3442e35c28ac2e4d6962b09286cad748f3c83d561b52ec6f</pre>
AWS PCS 代理程式 1.3.1-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.1-1.tar.gz</pre>	<pre>5b7f1eb7b3a86bd2d331b5cb0138d868dc9452da34b480becd86af892c7e8d19</pre>

Installer (安裝程式)	下載 URL	SHA256 檢查總和
AWS PCS 代理程式 1.3.0-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.0-1.tar.gz</pre>	<pre>eadc9b65c3db248bdd e2a6c41814dfb1b972 39f24ad55e03d8526d d9ab4a8d16</pre>
AWS PCS 代理程式 1.2.2-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.2-1.tar.gz</pre>	<pre>fd7b6ea5442db75d72 3fc4971781ce6ae511 baa21b87c4286fc1df 8127b282b8</pre>
AWS PCS 代理程式 1.2.1-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.1-1.tar.gz</pre>	<pre>2b784643ca01ccca1b aa64fbfb34bb41efe8 bdca69470998b74ce3 962bc271d4</pre>
AWS PCS 代理程式 1.2.0-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.0-1.tar.gz</pre>	<pre>470db8c4fc9e50277b 6317f98584b6b547e7 3523043e34f018eeca e767846805</pre>
AWS PCS 代理程式 1.1.1-1	<pre>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz</pre>	<pre>bef078bf60a6d8ecde 2e6c49cd34d088703f 02550279e3bf483d57 a235334dc6</pre>

Installer (安裝程式)	下載 URL	SHA256 檢查總和
AWS PCS 代理程式 1.1.0-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.0-1.tar.gz</code>	594c32194c71bccc5d66e5213213ae38dd2c6d2f9a950bb01acceae0bbab0873a
AWS PCS 代理程式 1.0.1-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.1-1.tar.gz</code>	04e22264019837e3f42d8346daf5886eaaced21571742eb505ea8911786bcb2
AWS PCS 代理程式 1.0.0-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz</code>	d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042144b134ce0

Slurm 安裝程式

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 25.05.5-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz</code>	e7bc84db4e71b8c7174e2f581a31233f839affb5306c76a8adba23204dcc703b
Slurm 25.05.4-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs</code>	3b0f93bce441d4f4f6935175f2c1e81cd961

Installer (安裝程式)	下載 URL	SHA256 檢查總和
	-slurm-25.05-installer-25.05.4-1.tar.gz	cb923adb416fa6689f5592047a7d
Slurm 25.05.3-1	https://aws-pcs-repo- <i>us-east-1</i> .s3. <i>us-east-1</i> .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.3-1.tar.gz	851bb5815b6700ceb30cc4a3fda204ca8ce362c14528c339908983255a936cf0
Slurm 24.11.7-1	https://aws-pcs-repo- <i>us-east-1</i> .s3. <i>us-east-1</i> .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.7-1.tar.gz	73d75be82c6f88f6e248fd0cc779a5630c62d91ebabdd9cf0f61b1943b6d7d09
Slurm 24.11.6-2	https://aws-pcs-repo- <i>us-east-1</i> .s3. <i>us-east-1</i> .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.6-2.tar.gz	f17cd78e0bc6b9c818b794d9d2685cceabdc73f4fbb12f7566ae5b86a5abc32b
Slurm 24.11.6-1	https://aws-pcs-repo- <i>us-east-1</i> .s3. <i>us-east-1</i> .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.6-1.tar.gz	225de9fc18206f5f65f412effe1fd457614ac97ee9822b3ff804a452b0fae522

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 24.11.5-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz</code>	593efe4d66bef2f3e46d5a382fb5a32f7a3ca2510bcf1b3c85739f4f951810d5
Slurm 24.05.8-2	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-2.tar.gz</code>	c494b0b55c319a4c2f3faf668c759d46c32c4c7aa94ae97d94128328fe95364b
Slurm 24.05.8-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-1.tar.gz</code>	210a43b376af082bbad640b2032655885790c5dab0e6489cc327c7310a375849
Slurm 24.05.7-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.7-1.tar.gz</code>	0b5ed7c81195de2628c78f37c79e63fc4ae99132ca6b019b53a0d68792ee82c5
Slurm 24.05.5-2	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz</code>	7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 23.11.10-4 (已棄用)	https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-4.tar.gz	bb2d8c919c69dba38d14358f49c7f0427564c5dd4af85a1c9eca2c57ceeae29a
Slurm 23.11.10-3 (已棄用)	https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-3.tar.gz	488a10ee0fbd57ec0e0ff7ea708a9e3038fafdc025c6bb391c75c2e2a7852a00
Slurm 23.11.10-2 (已棄用)	https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-2.tar.gz	0bbe85423305c05987931168caf98da08a34c25f9eec0690e8e74de0b7bc8752
Slurm 23.11.10-1 (已棄用)	https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-1.tar.gz	27e8faa9980e92cdfd8cfdc71f937777f0934552ce61e33dac4ecf5a20321e44

Installer (安裝程式)	下載 URL	SHA256 檢查總和
Slurm 23.11.9-1 (已棄用)	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz</code>	<code>1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8</code>

AWS PCS 範例 AMIs 版本備註

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

Important

不支援與舊排程器版本相關的範例 AMIs，也不會接收更新。

Important

範例 AMIs 僅供示範之用，不建議用於生產工作負載。

內容

- [AWS x86_64 的 PCS 範例 AMIs \(Amazon Linux 2\)](#)
- [AWS Arm64 \(Amazon Linux 2\) 的 PCS 範例 AMIs](#)

AWS x86_64 的 PCS 範例 AMIs (Amazon Linux 2)

Slurm 25.05

AMI 名稱

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05`

支援的 EC2 執行個體

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

AMI 內容

- 支援的 AWS 服務：AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構：x86_64
- EBS 磁碟區類型：gp2
- EFA 安裝程式：1.43.1
- GDRCopy：2.5.1
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

Slurm 24.11

Note

AWS PCS 支援 Slurm 24.11 和更新版本。如需詳細資訊，請參閱 [AWS PCS 中的 Slurm 會計](#)。

AMI 名稱

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

支援的 EC2 執行個體

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

AMI 內容

- 支援 AWS 的服務：AWS PCS

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

Slurm 24.05

AMI 名稱

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

支援的 EC2 執行個體

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

AMI 內容

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

Slurm 23.11

AMI 名稱

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

支援的 EC2 執行個體

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

AMI 內容

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

AWS Arm64 (Amazon Linux 2) 的 PCS 範例 AMIs

Slurm 25.05

AMI 名稱

- aws-pcs-sample_ami-amzn2-arm64-slurm-25.05

支援的 EC2 執行個體

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

AMI 內容

- 支援的 AWS 服務：AWS PCS
- 作業系統：Amazon Linux 2
- 運算架構：arm64
- EBS 磁碟區類型：gp2

- EFA 安裝程式：1.43.1
- GDRCopy：2.5.1
- NVIDIA 驅動程式：550.127.08
- NVIDIA CUDA：12.4.1_550.54.15

Slurm 24.11

Note

AWS PCS 支援 Slurm 24.11 和更新版本。如需詳細資訊，請參閱[AWS PCS 中的 Slurm 會計](#)。

AMI 名稱

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

支援的 EC2 執行個體

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

AMI 內容

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

Slurm 24.05

AMI 名稱

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

支援的 EC2 執行個體

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

AMI 內容

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

Slurm 23.11

AMI 名稱

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

支援的 EC2 執行個體

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

AMI 內容

- 支援 AWS 的服務：AWS PCS

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

AWS PCS 中支援的作業系統

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

支援的作業系統

- Amazon Linux 2

這是 AWS PCS 範例 AMIs 中的作業系統。

Important

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

- Amazon Linux 2023
- RedHat Enterprise Linux 9 (RHEL 9)

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

- RedHat Enterprise Linux 8 (RHEL 8)
- Rocky Linux 9

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

升級核心

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

```
sudo yum -y update
```

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

- Rocky Linux 8

- Ubuntu 22.04

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

- Ubuntu 24.04

AWS PCS 代理程式版本

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

AWS PCS 代理程式版本	版本日期	版本備註
1.3.2-1 版	2026 年 3 月 10 日	<ul style="list-style-type: none"> 修正執行 RHEL 8.10 或 Rocky Linux 8.10 的運算節點因為這些作業系統中的 curl SigV4 後端連接埠故障而無法引導的問題。
1.3.1-1 版	2025 年 11 月 7 日	<ul style="list-style-type: none"> 改善停用超執行緒的功能，方法是在可用時使用 `smt/control` sysfs 參數。 已修正在 PCS 代理程式嘗試停用超執行緒時，CPU 在開機期間鎖定時的潛在競爭條件。 已修正導致 Slurm 運算節點的 InstanceId 和 InstanceType 欄位分別填入時間戳記和連字號的問題。
1.3.0-1 版	2025 年 11 月 3 日	<ul style="list-style-type: none"> 新增對新作業系統的支援：Amazon Linux 2023、Ubuntu 24、RHEL 8、Rocky 8。

AWS PCS 代理程式版本	版本日期	版本備註
1.2.2-1 版	2025 年 10 月 16 日	<ul style="list-style-type: none"> • 如果 IPv4 端點無法使用，允許對 IPv6 端點的執行個體中繼資料查詢。IPv4 • 修正核心傳回同級執行緒做為 CPU ID 範圍時，無法停用超執行緒的問題。 • 修正在成功停用超執行緒時，日誌中產生錯誤失敗訊息的問題。
1.2.1-1 版	2025 年 6 月 19 日	<ul style="list-style-type: none"> • 如果控制器無法使用，AWS PCS 代理程式現在會嘗試啟動 slurmd 最多 30 分鐘。 • 修正以下問題：如果對 RegisterComputeNodeGroupInstance 的回應包含 SLURMDBD 端點，則會產生不正確的 slurmd 組態。
1.2.0-1 版	2025 年 3 月 7 日	<ul style="list-style-type: none"> • 在 <code>slurmd.conf</code> 中啟用 IPv6 支援。
1.1.1-1 版	2024 年 12 月 13 日	<ul style="list-style-type: none"> • 修正呼叫 RegisterComputeNodeGroupInstance 時回報不正確 Slurm 版本的問題。 • 修正在 <code>/opt/aws/pcs/etc/bootstrap_hooks/</code> 執行自訂指令碼時，未正確擷取執行個體中繼資料的問題。

AWS PCS 代理程式版本	版本日期	版本備註
1.1.0-1 版	2024 年 12 月 6 日	<ul style="list-style-type: none">• 啟用 中的自訂指令碼/<code>opt/aws/pcs/etc/bootstrap_hooks/</code>，以在引導步驟之前執行。
1.0.1-1 版	2024 年 10 月 22 日	<ul style="list-style-type: none">• 修正 NVIDIA 裝置在啟用 GPU 的執行個體上 <code>slurmd</code> 啟動時無法運作的問題。
1.0.0-1 版	2024 年 8 月 28 日	<ul style="list-style-type: none">• 初始版本。

AWS PCS 中的 Slurm 排程器

Slurm 是專為 Linux 叢集設計的開放原始碼工作負載管理器，可為 HPC 工作負載提供任務排程、資源配置和任務監控功能。AWS PCS 支援 Slurm 排程器來管理您的叢集工作負載。

主題

- [AWS PCS 中的 Slurm 版本](#)
- [AWS PCS 中的 Slurm 會計](#)
- [AWS PCS 中的 Slurm REST API](#)
- [在 AWS PCS 中使用 Slurm 重新啟動運算節點](#)
- [在 AWS PCS 中設定自訂 Slurm 設定](#)
- [在 AWS PCS 中設定自訂 cgroup 設定](#)
- [在 AWS PCS 中設定自訂 SlurmDBD 設定](#)
- [使用 SPANK 外掛程式擴展 AWS PCS 上的 Slurm 功能](#)
- [使用 Slurm CLI 篩選條件外掛程式在 AWS PCS 中自訂任務提交](#)

AWS PCS 中的 Slurm 版本

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

當 SchedMD 結束[支援](#)特定主要版本時，AWS PCS 會將該版本指定為生命週期結束 (EOL)。在 EOL 之後，無法使用該版本建立新的叢集，但現有叢集可以在沒有保證支援的情況下繼續執行長達 12 個月。如果 Slurm 主要版本接近 EOL，AWS PCS 會傳送預先通知，協助客戶知道何時將其叢集升級到較新的支援版本。

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

AWS PCS 中支援的 Slurm 版本

下表顯示支援的 Slurm 版本，以及每個版本的重要日期和資訊。

Slurm 版本	SchedMD 發行日期	AWS PCS 發行日期	AWS PCS EOL 日期	最低相容 AWS PCS 代理程式版本	支援的 AWS PCS 範例 AMIs
25.05	5/29/2025	10/16/2025	11/30/2026	1.0.0-1	<ul style="list-style-type: none"> aws-pcs-s ample_ami -amzn2-x86_64-slurm-25.05 aws-pcs-s ample_ami -amzn2-arm64-slurm-25.05
24.11	11/29/2024	5/14/2025	5/31/2026	1.0.0-1	<ul style="list-style-type: none"> aws-pcs-s ample_ami -amzn2-x86_64-slurm-24.11 aws-pcs-s ample_ami -amzn2-arm64-slurm-24.11

AWS PCS 中不支援的 Slurm 版本

下表顯示 AWS PCS 中不支援的 Slurm 版本。

Slurm 版本	SchedMD 發行日期	AWS PCS 發行日期	AWS PCS EOL 日期		
24.05	5/30/2024	12/18/2024	11/30/2025		
23.11	11/21/2023	8/28/2024	5/31/2025		

AWS PCS 中 Slurm 版本的版本備註

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

Slurm 25.05

在 AWS PCS 中實作的變更

- Slurm requeue_on_resume_failure SchedulerParameter 現在預設為啟用。
- 已移除「stderr」做為 LogTimeFormat 的選項，因為它已在 Slurm 25.05 中停用。
- AWS PCS 支援多叢集堆疊組態：登入節點可以存取多個叢集。

如需 Slurm 25.05 的詳細資訊，請參閱下列出版物：

- SchedMD 版本公告：<https://www.schedmd.com/slurm-version-25-05-0-is-now-available/>
- SchedMD 版本備註：https://github.com/SchedMD/slurm/blob/slurm-25-05-0-1/RELEASE_NOTES.md

Slurm 24.11

在 AWS PCS 中實作的變更

- AWS PCS 支援 Slurm 會計。如需詳細資訊，請參閱[AWS PCS 中的 Slurm 會計](#)。

如需 Slurm 24.11 的詳細資訊，請參閱下列出版物：

- [SchedMD 版本公告](#)
- [SchedMD 版本備註](#)

Slurm 24.05

在 AWS PCS 中實作的變更

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

如需 Slurm 24.05 的詳細資訊，請參閱下列出版物：

- [SchedMD 版本公告](#)
- [SchedMD 版本備註](#)

Slurm 23.11

您可以在 AWS PCS 中變更 Slurm 設定

- SuspendTime 預設為 60。使用 AWS PCS scaleDownIdleTimeInSeconds組態參數進行設定。如需詳細資訊，請參閱 AWS PCS API 參考中 ClusterSlurmConfiguration 資料類型的 [scaleDownIdleTimeInSeconds](#) 參數。
- MaxJobCount 和 MaxArraySize是根據您為叢集選擇的大小。如需詳細資訊，請參閱 PCS CreateCluster API 參考中 API 動作的 [size](#) 參數。AWS
- SelectTypeParameters Slurm 設定預設為 CR_CPU。您可以提供它做為值slurmCustomSettings，讓在建立叢集時設定它。如需詳細資訊，請參閱 AWS PCS API 參考中 CreateCluster API 動作和 [SlurmCustomSetting](#) 的 [slurmCustomSettings](#) 參數。

- 您可以在叢集層級設定 `Epilog Prolog` 和 `RealMemory Weight`。您可以提供它做為值 `slurmCustomSettings`，讓在建立叢集時設定它。如需詳細資訊，請參閱 AWS PCS API 參考中的 [CreateCluster](#) 和 [SlurmCustomSetting](#)。
- 您可以在運算節點群組層級設定 `RealMemory Weight` 和 `RealMemory Weight`。您可以在建立運算節點群組時，提供它做為的值 `slurmCustomSettings`。如需詳細資訊，請參閱 AWS PCS API 參考中的 [CreateComputeNodeGroup](#) 和 [SlurmCustomSetting](#)。

有關 AWS PCS 中 Slurm 版本的常見問題

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

AWS PCS 支援 Slurm 版本多久？

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

我的叢集如何取得新的 Slurm 修補程式版本？

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

Note

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

如何通知即將發生的 Slurm 版本 EOL 事件？

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

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

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

如果我的 Slurm 版本接近或超過 EOL，我需要做什麼？

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

如果我未在 EOL 日期之前切換到較新版本的 Slurm，會發生什麼情況？

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

使用 EOL Slurm 版本操作叢集有哪些風險？

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

叢集暫停時，我的任務、叢集運算、儲存和聯網資源會發生什麼情況？

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

我可以重新啟動暫停的叢集以繼續其剩餘的任務嗎？

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

我可以請求超過 12 個月寬限期的延長嗎？

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

AWS PCS 中的 Slurm 會計

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

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

Note

Slurm 24.11 或更新版本支援會計。

AWS PCS console

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

AWS PCS API

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

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

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

Important

如果您啟用會計，則會收到額外的帳單費用。如需詳細資訊，請參閱 [AWS PCS 定價頁面](#)。

修改會計設定

您可以在現有叢集上啟用或停用會計，而無需重建您的基礎設施。如需詳細資訊，請參閱在 [AWS PCS 中更新叢集](#)。

當您停用會計時，會計功能的帳單會在叢集進入 UPDATING 狀態後立即停止。當您啟用會計時，帳單會在叢集成功返回 ACTIVE 狀態時開始。

AWS PCS 中 Slurm 會計的重要概念

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

會計資料庫

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

預設清除時間

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

AWS PCS console

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

AWS PCS API

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

Note

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

會計政策強制執行

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

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

如需這些選項的詳細資訊，請參閱 [SchedMD 中的 Slurm 文件](#)。

AWS PCS console

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

AWS PCS API

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

取得現有 AWS PCS 叢集的會計組態

Slurm 會計組態包含在叢集的 Slurm 組態中。

AWS PCS console

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

AWS PCS API

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

AWS PCS 中的 Slurm REST API

AWS PCS 透過 `slurmrestd` 為 Slurm 的原生 REST API 提供受管支援，為程式設計叢集互動提供 HTTP 介面。您可以透過標準 HTTP 請求提交任務、監控叢集狀態和管理資源，而不需要直接透過 shell 存取叢集。

常用案例

Slurm REST API 支援各種整合案例：

- Web 應用程式整合：建置自訂前端和 Web 應用程式，以直接提交和管理任務。
- Jupyter 筆記本整合：允許研究人員從筆記本環境提交任務，而無需離開其開發工作流程。
- 合作夥伴解決方案整合：將第三方 HPC 工具和工作流程管理員連接到 AWS PCS 叢集。
- 程式設計叢集管理：自動化任務提交、監控和資源管理工作流程。
- 研究運算工作流程：支援需要 API 驅動型任務管理的學術和企業研究環境。

要求與限制

使用 Slurm REST API 之前，請檢閱下列詳細資訊：

- 您的叢集必須使用 Slurm 25.05 版或更新版本。
- API 端點只能透過叢集 VPC 內的私有 IP 地址存取。
- 您的叢集安全群組必須允許連接埠 6820 上的 HTTP 流量。

- 身分驗證需要具有特定使用者身分宣告的 JWT 字符。

目前的限制包括：

- `scontrol` token 不支援產生的字符。
- `X-SLURM-USER-NAME` 標頭模擬不可用。
- 某些功能需要啟用 Slurm 會計。
- 與 Slurm CLI 篩選條件外掛程式機制不相容。
- REST API 端點的連線不會使用 TLS 加密。

主題

- [在 AWS PCS 中啟用 Slurm REST API](#)
- [在 AWS PCS 中使用 Slurm REST API 驗證](#)
- [在 AWS PCS 中使用 Slurm REST API 進行任務管理](#)
- [AWS PCS 中的 Slurm REST API 常見問答集](#)

在 AWS PCS 中啟用 Slurm REST API

啟用 Slurm REST API 來存取叢集的 HTTP 介面，以進行程式設計任務管理和監控。您可以在叢集建立期間啟用此功能，或更新符合需求的現有叢集。

先決條件

啟用 Slurm REST API 之前，請確定您已：

- 叢集版本：Slurm 25.05 版或更新版本。
- 安全群組：允許連接埠 6820 上來自所需來源的 HTTP 流量的規則。

程序

在新叢集上啟用 Slurm REST API

AWS 管理主控台

1. 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
2. 選擇 Create Cluster (建立叢集)。

3. 在叢集詳細資訊下，選擇 Slurm 25.05 版或更新版本。
4. 視需要設定其他叢集設定。
5. 在排程器組態區段中，將 REST API 設定為已啟用。
6. 設定叢集安全群組，以允許連接埠 6820 上來自所需來源的 HTTP 流量。
7. 完成叢集建立程序。

AWS CLI

1. 在建立叢集時新增 Slurm REST 組態。

```
aws pcs create-cluster --region region \  
  --cluster-name my-cluster \  
  --scheduler type=SLURM, version=25.05 \  
  --size SMALL \  
  --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1 \  
  --slurm-configuration slurmRest='{mode=STANDARD}'
```

2. 設定叢集安全群組，以允許連接埠 6820 上來自所需來源的 HTTP 流量。

在現有叢集上啟用 Slurm REST API

AWS 管理主控台

1. 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
2. 從清單中選擇您的叢集。
3. 確認您的叢集在叢集詳細資訊中使用 Slurm 25.05 版或更新版本。
4. 選擇編輯叢集。
5. 在排程器組態區段中，將 REST API 設定為已啟用。
6. 選擇更新叢集以套用變更。
7. 設定叢集安全群組，以允許連接埠 6820 上來自所需來源的 HTTP 流量。

AWS CLI

1. 使用 Slurm REST 組態更新您的叢集，如本範例所示。

```
aws pcs update-cluster --cluster-identifier my-cluster \  
  --slurm-configuration slurmRest='{mode=STANDARD}'
```

```
--slurm-configuration 'slurmRest={mode=STANDARD}'
```

2. 設定叢集安全群組，以允許連接埠 6820 上來自所需來源的 HTTP 流量。

啟用後會發生什麼情況

當您啟用 REST API 時，AWS PCS 會自動：

- 產生 JWT 簽署金鑰並將其存放在 AWS Secrets Manager 中。
- 在 VPC `https://<clusterPrivateIpAddress>:6820` 中公開 API 端點。
- 更新您的叢集組態，以顯示 REST API 端點詳細資訊。

您現在可以驗證和使用 REST API 進行任務管理和叢集操作。

在 AWS PCS 中使用 Slurm REST API 驗證

AWS PCS 中的 Slurm REST API 使用 JSON Web Token (JWT) 身分驗證來確保對叢集資源的安全存取。AWS PCS 提供存放在 AWS Secrets Manager 中的受管簽署金鑰，您可用來產生包含特定使用者身分宣告的 JWT 權杖。

先決條件

使用 Slurm REST API 驗證之前，請確定您有：

- 叢集組態：已啟用 Slurm 25.05+ 和 REST API 的 AWS PCS 叢集。
- AWS 許可：存取 JWT 簽署金鑰的 AWS Secrets Manager。
- 使用者資訊：您叢集帳戶的使用者名稱、POSIX 使用者 ID 和一或多個 POSIX 群組 IDs。
- 網路存取：叢集 VPC 內的連線與允許連接埠 6820 的安全群組。

程序

擷取 Slurm REST API 端點地址

AWS 管理主控台

1. 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
2. 從清單中選擇您的叢集。
3. 在叢集組態詳細資訊中，找到端點區段。

4. 請注意 Slurm REST API (slurmrestd) 的私有 IP 地址和連接埠。
5. 您可以透過傳送格式正確的 HTTP 請求到此地址來進行 API 呼叫。

AWS CLI

1. 使用查詢您的叢集狀態 `aws pcs get-cluster`。在回應的 `endpoints` 欄位中尋找 SLURMRESTD 端點。請見此處範例：

```
"endpoints": [  
  {  
    "type": "SLURMCTLD",  
    "privateIpAddress": "192.0.2.1",  
    "port": "6817"  
  },  
  {  
    "type": "SLURMRESTD",  
    "privateIpAddress": "192.0.2.1",  
    "port": "6820"  
  }  
]
```

2. 您可以將格式正確的 HTTP 請求傳送至 `http://<privateIpAddress>:<port>/`

擷取 JWT 簽署金鑰

1. 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
2. 從清單中選擇您的叢集。
3. 在叢集組態詳細資訊中，找到排程器身分驗證區段。
4. 請注意 JSON Web Token (JWT) 金鑰 ARN 和版本。
5. 使用從 Secrets Manager AWS CLI 擷取簽署金鑰：

```
aws secretsmanager get-secret-value --secret-id arn:aws:secretsmanager:region:account:secret:name --version-id version
```

產生 JWT 字符

1. 使用下列必要宣告建立 JWT：

- `exp` – JWT 自 1970 以來的過期時間，以秒為單位
 - `iat` – 自 1970 以來的目前時間，以秒為單位
 - `sub` – 身分驗證的使用者名稱
 - `uid` – POSIX 使用者 ID
 - `gid` – POSIX 群組 ID
 - `id` – 其他 POSIX 身分屬性
 - `gecos` – 使用者註解欄位，通常用於存放人類可讀取的名稱
 - `dir` – 使用者的主目錄
 - `shell` – 使用者的預設 shell
 - `gids` – 使用者所在的其他 POSIX 群組 IDs 清單
2. 使用從 Secrets Manager 擷取的簽署金鑰簽署 JWT。
 3. 為字串設定適當的過期時間。

Note

作為 `sub` 宣告的替代方案，您可以提供下列任何一項：

- `username`
- 您在 `userclaimfield` 中透過定義的自訂欄位名稱 `AuthAltParameters Slurm custom settings`
- `id` 宣告中的 `name` 欄位

驗證 API 請求

1. 使用下列其中一種方法在 HTTP 請求中包含 JWT 字串：
 - 承載字串 – 新增 `Authorization: Bearer <jwt>` 標頭
 - Slurm 標頭 – 新增 `X-SLURM-USER-TOKEN: <jwt>` 標頭
2. 向 REST API 端點提出 HTTP 請求：

以下是使用 `curl` 和 `Authorization: Bearer` 標頭存取 `/ping` API 的範例。

```
curl -X GET -H "Authorization: Bearer <jwt>" \
```

```
http://<privateIpAddress>:6820/slurm/v0.0.43/ping
```

產生 JWT 的範例

擷取 AWS PCS 叢集 JWT 簽署金鑰，並將其儲存為本機檔案。將 `aws-region`、`secret-arn` 和 `secret` 版本的值取代為您的叢集適用的值。

```
#!/bin/bash
SECRET_KEY=$(aws secretsmanager get-secret-value \
  --region aws-region \
  --secret-id secret-arn \
  --version-stage secret-version \
  --query 'SecretString' \
  --output text)
echo "$SECRET_KEY" | base64 --decode > jwt.key
```

此 Python 範例說明如何使用簽署金鑰來產生 JWT 字符：

```
#!/usr/bin/env python3

import sys
import os
import pprint
import json
import time
from datetime import datetime, timedelta, timezone
from jwt import JWT
from jwt.jwa import HS256
from jwt.jwk import jwk_from_dict
from jwt.utils import b64decode, b64encode
if len(sys.argv) != 3:
    sys.exit("Usage: gen_jwt.py [jwt_key_file] [expiration_time_seconds]")
SIGNING_KEY = sys.argv[1]
EXPIRATION_TIME = int(sys.argv[2])
with open(SIGNING_KEY, "rb") as f:
    priv_key = f.read()
signing_key = jwk_from_dict({
    'kty': 'oct',
    'k': b64encode(priv_key)
})
message = {
    "exp": int(time.time() + EXPIRATION_TIME),
```

```
"iat": int(time.time()),
"sun": "ec2-user",
"uid": 1000,
"gid": 1000,
"id": {
    "gecos": "EC2 User",
    "dir": "/home/ec2-user",
    "gids": [1000],
    "shell": "/bin/bash"
}
}
a = JWT()
compact_jws = a.encode(message, signing_key, alg='HS256')
print(compact_jws)
```

指令碼會將 JWT 列印到螢幕。

```
abcdefghijklmnopjwttoken...
```

在 AWS PCS 中使用 Slurm REST API 進行任務管理

Slurm REST API 概觀

Slurm REST API 可透過 HTTP 請求，以程式設計方式存取叢集管理函數。了解這些關鍵特性可協助您有效地將 API 與 AWS PCS 搭配使用：

- 存取通訊協定：API 使用 HTTP (而非 HTTPS) 進行叢集私有網路內的通訊。
- 連線詳細資訊：使用叢集的私有 IP 地址和 `slurmrestd` 連接埠 (通常是 6820) 存取 API。完整的基本 URL 格式為 `http://<privateIpAddress>:6820`。
- API 版本控制：API 版本對應於您的 Slurm 安裝。對於 Slurm 25.05，請使用版本 v0.0.43。版本編號會隨每個 Slurm 版本而變更。您可以在 [Slurm 版本備註中找到目前支援的 API 版本](#)。
- URL 結構：Slurm REST API 的 URL 結構為 `http://<privateIpAddress>:<port>/<api-version>/<endpoint>`。您可以在 [Slurm 文件](#) 中找到 REST API 端點的詳細用量資訊。

如需使用 Slurm REST API 的特定資訊，請參閱 REST 用戶端 [Slurm 文件](#)。

先決條件

使用 Slurm REST API 之前，請確定您已：

- 叢集組態：已啟用 Slurm 25.05+ 和 REST API 的 AWS PCS 叢集。
- 身分驗證：具有適當使用者身分宣告的有效 JWT 字符。
- 網路存取：叢集 VPC 內的連線，以及允許連接埠 6820 的安全群組。

程序

使用 REST API 提交任務

1. 使用必要的參數建立任務提交請求：

```
{
  "job": {
    "name": "my-job",
    "partition": "compute",
    "nodes": 1,
    "tasks": 1,
    "script": "#!/bin/bash\nnecho 'Hello from Slurm REST API'",
    "environment": ["PATH=/usr/local/bin:/usr/bin:/bin"]
  }
}
```

2. 使用 HTTP POST 請求提交任務：

```
curl -X POST \
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '<job-json>' \
  https://<privateIpAddress>:6820/slurm/v0.0.43/job/submit
```

3. 請注意回應中傳回的任務 ID，以用於監控。

監控任務狀態

1. 取得特定任務的相關資訊：

```
curl -X GET -H "Authorization: Bearer <jwt>" \
  https://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

2. 列出已驗證使用者的所有任務：

```
curl -X GET -H "Authorization: Bearer <jwt>" \  
https://<privateIpAddress>:6820/slurm/v0.0.43/jobs
```

取消任務

- 傳送 DELETE 請求以取消特定任務：

```
curl -X DELETE -H "Authorization: Bearer <jwt>" \  
https://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

AWS PCS 中的 Slurm REST API 常見問答集

本節回答有關 AWS PCS 中 Slurm REST API 的常見問題。

什麼是 Slurm REST API？

Slurm REST API 是一種 HTTP 介面，可讓您以程式設計方式與 Slurm 工作負載管理員互動。您可以使用 GET、POST 和 DELETE 等標準 HTTP 方法來提交任務、監控叢集狀態和管理資源，而無需對叢集進行命令列存取。

我可以從產生的權杖 **scontrol token** 嗎？

否，標準 **scontrol token** 輸出與 AWS PCS 不相容。PCS Slurm REST API 需要包含特定身分宣告的豐富 JWT 字符，包括 `username(sun)`、POSIX 使用者 ID(`uid`) 和群組 IDs(`gids`)。標準 Slurm 權杖缺少這些必要的宣告，且會被 API 拒絕。

我可以從 VPC 外部存取 API 嗎？

否，REST API 端點只能使用 Slurm 控制器的私有 IP 地址從 VPC 內存取。若要啟用外部存取，請使用 VPC Link、API Gateway 實作 Application Load Balancer 等 AWS 服務，或建立 VPC 對等互連或 VPN 連線以實現安全連線。

為什麼 API 使用 HTTP 而不是 HTTPS？

Slurm REST API 旨在成為叢集私有網路內的內部端點。對於需要加密的生產部署，您可以在架構的更高層級實作 SSL/TLS 終止，例如透過 API 閘道、負載平衡器或反向代理。

如何控制對 REST API 的存取？

設定叢集的安全群組規則，以限制對 Slurm 控制器上連接埠 6820 的存取。設定傳入規則以僅允許來自信任 IP 範圍或 VPC 內特定來源的連線，封鎖對 API 端點的未經授權存取。

如何輪換 JWT 簽署金鑰？

將您的叢集置於沒有作用中執行個體的維護模式中，然後透過 AWS Secrets Manager 啟動金鑰輪換。輪換完成後，重新啟用佇列。所有現有的 JWT 字符都將失效，並且必須使用 Secrets Manager 的新簽署金鑰重新產生。

我需要啟用 Slurm 會計才能使用 REST API 嗎？

否，基本 REST API 操作不需要 Slurm 會計，例如任務提交和監控。不過，整個/slurmdb端點需要處於作用中狀態。

哪些第三方工具可與 AWS PCS REST API 搭配使用？

許多現有的 Slurm REST API 用戶端應該使用 AWS PCS，包括適用於 Prometheus 的 Slurm Exporter、SlurmWeb，以及遵循標準 Slurm REST API 格式的自訂應用程式。不過，依賴 scontrol token 進行身分驗證的工具將需要修改，才能使用 AWS PCS JWT 要求。

使用 REST API 是否有任何額外費用？

否，啟用或停用 Slurm REST API 功能無需額外費用。您只需照常支付基礎叢集資源的費用。

如何對 REST API 進行故障診斷？

• 網路連線問題

如果您無法連線到 API 端點，在向叢集控制器提出 HTTP 請求時，您會看到連線逾時或「連線遭拒」錯誤。

做法如下：確認您的用戶端位於相同的 VPC 中或具有適當的網路路由，並確認安全群組允許來自來源 IP 或子網路的連接埠 6820 上的 HTTP 流量。

• Slurm REST 身分驗證問題

如果您的 JWT 權杖無效、過期或簽署不當，API 請求會在回應的錯誤欄位中傳回「通訊協定身分驗證錯誤」。

錯誤訊息範例：

```
{
  "errors": [
    {
      "description": "Batch job submission failed",
      "error_number": 1007,
      "error": "Protocol authentication error",
      "source": "slurm_submit_batch_job()"
    }
  ]
}
```

```
]
}
```

做法：檢查您的 JWT 權杖格式正確、未過期，並使用 Secrets Manager 的正確金鑰進行簽署。確認字符格式正確，並包含必要的宣告，而且您使用的是正確的身分驗證標頭格式。

- 提交後任務無法執行

如果您的 JWT 字符有效，但包含不正確的內部結構或內容，則任務可能已進入暫停 (PD) 狀態，原因碼為 JobAdminHead。使用 `scontrol show job <job-id>` 來檢查任務 – 您將看到 JobState=PENDING, Reason=JobHeldAdmin、和 SystemComment=slurm_cred_create failure, holding job。

做法：根本原因可能是 JWT 中的錯誤值。確認字符結構正確，並根據 PCS 文件包含必要的宣告。

- 工作目錄許可問題

如果 JWT 中指定的使用者身分缺少對任務工作目錄的寫入許可，任務將失敗並出現許可錯誤，類似於使用 `sbatch --chdir` 搭配無法存取的目錄。

做法：確保 JWT 權杖中指定的使用者具有任務工作目錄的適當許可。

- 仍然遇到問題？

1. 檢查 REST API 規格上的 SchedMD [文件](#)。
2. 如需錯誤的詳細資訊，請參閱 Slurm 控制器日誌（如需詳細資訊 [AWS PCS 中的排程器日誌](#)，請參閱）。

在 AWS PCS 中使用 Slurm 重新啟動運算節點

AWS PCS 支援 Slurm 的原生 `scontrol reboot` 命令。使用此命令重新啟動運算節點，無需更換 EC2 執行個體。其他重新啟動方法 (Amazon EC2 主控台、AWS CLI 自動化修補程式或系統維護) 會導致 AWS PCS 認為 EC2 執行個體運作狀態不佳並予以取代。

Slurm 重新開機的優點

Slurm 重新開機為叢集維護提供了幾個優點：

- 保留容量 – 避免將容量受限的 EC2 執行個體遺失給其他客戶。
- 降低成本 – 消除不必要的執行個體替換週期，並持續為閒置節點計費。

- 更快的復原 – 相較於執行個體替換，佈建不會延遲。
- 操作彈性 – 清除記憶體流失、移除暫存檔案，以及從降級狀態復原節點。

何時使用 Slurm 重新開機

針對常見的操作維護案例使用 Slurm 重新啟動：

- 故障診斷 – 解決效能問題或無回應的程序，特別是 GPU 節點。
- 資源清除 – 清除記憶體流失、中的暫存檔案/tmp，或影響任務效能的卡住程序。
- 復原 – 在需要完全替換節點之前，從停止或降級狀態復原節點。

限制

- 只有 Slurm Admin 使用者（根使用者）可以執行重新啟動命令。
- 重新啟動支援僅限於 `scontrol reboot`。
- 不支援 `RebootProgram` 組態。
- 無主控台界面 – 僅限命令列。

主題

- [在 AWS PCS 中使用 Slurm 重新啟動運算節點](#)
- [在 AWS PCS 中取消待定重新啟動](#)
- [AWS PCS 中的 Slurm 重新啟動常見問答集](#)
- [針對 AWS PCS 中的 Slurm 重新啟動問題進行故障診斷](#)

在 AWS PCS 中使用 Slurm 重新啟動運算節點

使用 Slurm 的原生重新開機命令來解決效能問題、清除資源問題，或從降級狀態復原，而不會損失 EC2 執行個體容量。

先決條件

- Slurm Admin 權限（根使用者存取）
- 存取 AWS PCS 叢集中的登入節點

程序

1. 透過 EC2 主控台連線至登入節點。
 - a. 在 EC2 主控台中，選擇 Instances (執行個體)。
 - b. 選取您的登入節點執行個體。
 - c. 選擇連線。
2. 使用 `sinfo` 或 識別目標運算節點名稱 `scontrol show node`。

```
sinfo
# or
scontrol show node
```

3. 使用以下其中一個選項執行重新啟動命令：

Warning

請勿將 `nextstate=DOWN` 與 `scontrol reboot` 命令搭配使用。此參數會將節點標記為運作狀態不佳，並觸發執行個體替換。

- 基本重新啟動（等待節點變成閒置）：

```
scontrol reboot nodename
```

- 立即重新啟動（在任務完成時清空節點並重新啟動）：

```
scontrol reboot ASAP nodename
```

- 以下列原因重新啟動：

```
scontrol reboot ASAP reason="troubleshooting" nodename
```

- 以恢復狀態重新啟動：

```
scontrol reboot ASAP nextstate=RESUME nodename
```

4. 使用 監控重新啟動進度 `scontrol show node`。

```
scontrol show node nodename
```

5. 驗證節點在重新啟動完成後返回服務。

在 AWS PCS 中取消待定重新啟動

取消待重新開機，以避免問題解決或不再需要重新開機時不必要的停機時間。

先決條件

- Slurm Admin 權限
- 節點必須具有待重新啟動（顯示「重新啟動已發出」狀態）
- 存取登入節點以執行命令

程序

1. 連線至登入節點。
2. 使用 確認節點有待重新開機 `scontrol show node`。

```
scontrol show node nodename
```

尋找節點狀態的「已發出重新啟動」。

3. 執行取消命令。

```
scontrol cancel_reboot nodename
```

4. 驗證重新啟動取消和節點狀態是否恢復正常。

```
scontrol show node nodename
```

AWS PCS 中的 Slurm 重新啟動常見問答集

尋找有關在 AWS PCS 中使用 Slurm 重新啟動的常見問題解答。

什麼是 Slurm 重新啟動支援？

支援原生 Slurm `scontrol reboot` 命令。使用此命令無需自動替換執行個體即可重新啟動運算節點，以保留 EC2 執行個體容量並降低營運成本。

誰可以使用 Slurm 重新啟動命令？

只有 Slurm Admin 使用者（根使用者）可以執行重新啟動命令。嘗試使用的一般使用者 `scontrol reboot` 會收到來自 Slurm 的拒絕許可錯誤，而不會影響節點。

在重新開機期間執行任務會發生什麼情況？

根據預設，任務會在重新啟動之前正常完成。使用 ASAP 選項時，節點會耗盡以防止新任務，並在目前任務完成後重新啟動。任務可以取消或重新排入佇列，以便立即重新啟動。

這與 EC2 主控台重新啟動有何不同？

Slurm 重新啟動會保留 EC2 執行個體並避免取代，而 EC2 主控台重新啟動會觸發 PCS 因為重新啟動程序期間運作狀態檢查失敗而取代執行個體。

我可以設定自訂重新開機指令碼嗎？

否，初始版本不支援 `RebootProgram` 組態。此功能使用標準 Slurm 重新啟動行為，無需自訂指令碼支援。

Slurm 重新啟動需要多長時間？

重新啟動時間會根據執行個體類型、客戶開機程序、AMI 組態，以及任務是否需要先完成而有所不同。此程序包括等待任務完成、實體重新開機、運作狀態檢查和 `slurmd` 協助程式註冊。

我可以看到重新啟動的歷史記錄嗎？

重新啟動事件會記錄在可透過 CloudWatch 監控的 Slurm 日誌 (`slurmctld` 和 `slurmd`) 中。節點狀態中的原因欄位會顯示程序期間的重新啟動原因。

如果節點在重新開機期間卡住該怎麼辦？

如果節點未在 `ResumeTimeout` 中完成重新啟動程序，則會將其標示為 DOWN。檢查 CloudWatch 日誌是否有錯誤、驗證網路連線，以及檢查 `slurmd` 日誌。如果問題持續發生，請聯絡 AWS Support。

我可以一次重新啟動多個節點嗎？

是，您可以在重新啟動命令中指定多個節點：

```
scontrol reboot ASAP node1,node2,node3
```

如何重新啟動節點而不等待任務完成？

對於遇到影響多節點任務、顯著效能降低或 GPU 行為不穩定等問題時立即重新啟動節點，您有兩個選項：

- 取消和重新啟動 – 首先，使用取消受影響的任務 `scancel <job_id>`，然後使用 啟動立即重新啟動 `scontrol reboot ASAP <nodename>`。執行中的任務將終止，且需要在節點復原後重新提交。
- 耗盡和排入佇列（影響較小） – 首先啟動耗盡並使用重新啟動 `scontrol reboot ASAP <nodename>`，然後使用重新排入受影響任務的佇列 `scontrol requeue <job_id>`。這會讓任務回到待定狀態，而不是取消任務。

如果我指定 `nextstate=DOWN`，會發生什麼情況？

如果您指定 `nextstate=DOWN`，節點會在重新啟動並觸發執行個體取代後標示為運作狀態不佳。若要避免執行個體替換，請勿指定 `nextstate` 或使用 `nextstate=RESUME`。

其他資源

- 如需基本重新啟動程序，請參閱 [在 AWS PCS 中使用 Slurm 重新啟動運算節點](#)。
- 如需重新開機問題的疑難排解，請參閱 [針對 AWS PCS 中的 Slurm 重新啟動問題進行故障診斷](#)。
- 如需 Slurm 重新啟動文件，請參閱 [Slurm scontrol 文件](#)。

針對 AWS PCS 中的 Slurm 重新啟動問題進行故障診斷

當您遇到節點重新啟動問題時，請先使用 檢查節點狀態 `scontrol show node nodename`。然後檢查 Slurm (`slurmctld` 和 `slurmd`) 和系統日誌的 CloudWatch 日誌，以識別潛在的錯誤。

如需基本故障診斷，請驗證網路連線、檢查安全群組設定，並確保所有必要的服務在重新開機後執行。如果問題在基本疑難排解步驟後仍存在，請聯絡 AWS Support。聯絡支援時，請提供相關的日誌摘錄、節點狀態資訊，以及重新啟動嘗試的時間表，以協助加速解決程序。

其他資源

- 如需使用 CloudWatch 監控 AWS PCS 執行個體，請參閱 [使用 Amazon CloudWatch 監控 AWS PCS 執行個體](#)。
- 如需一般故障診斷，請參閱 [平行 AWS 運算服務中的問題故障診斷](#)。
- 如需 Slurm 文件，請參閱 [Slurm 故障診斷指南](#)。

在 AWS PCS 中設定自訂 Slurm 設定

使用自訂 Slurm 設定來設定叢集、佇列和運算節點群組資源的其他 Slurm 參數。此版本新增對佇列資源上 Slurm 設定的支援，提供對分割區特定行為的精細控制。

自訂 Slurm 設定的優點

自訂 Slurm 設定可精密控制 AWS PCS 型 HPC 環境。您可以實作詳細的會計、強制執行存取控制，並透過 quality-of-service 組態和先佔政策來最佳化工作負載執行。這些功能可確保關鍵任務接收必要的資源，同時維持有效率的叢集使用率。無論您管理 GPU 加速的工作負載、實作公平共用排程或控制任務生命週期，自訂設定都有助於讓您的 HPC 基礎設施符合營運需求和研究目標。

設定自訂設定

自訂 Slurm 設定可以在資源建立期間透過 AWS 主控台、CLI 或 SDKs 進行設定，或稍後透過更新操作進行修改。

AWS 管理主控台

針對任何資源類型（叢集、佇列或運算節點群組），導覽至建立或編輯頁面中的其他排程器設定。

新增設定

1. 選擇新增設定。
2. 從下拉式清單中選取參數名稱（其中包含簡短參數說明）。
3. 提供對應的值。

取消設定自訂設定

1. 選擇相關參數/值對旁的移除。
2. 建立或更新 資源。

AWS CLI

若要以程式設計方式管理自訂設定，請在建立或更新操作中使用 `SlurmCustomSettings` 欄位。

Example– 更新叢集上的 Prolog 參數

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  

```

```
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/prolog.sh"}]'
```

Example– 將佇列設定為叢集Default上的

```
aws pcs update-queue \  
  --cluster-identifier my-cluster \  
  --queue-identifier my-queue \  
  --slurm-configuration \  
  'SlurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Example– 在運算節點群組Features上設定自訂

```
aws pcs update-compute-node-group \  
  --cluster-identifier my-cluster \  
  --compute-node-group-identifier my-cng-1 \  
  --slurm-configuration \  
  'SlurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

驗證和錯誤處理

AWS PCS 實作自訂 Slurm 設定的多層驗證程序。在建立和更新操作期間，我們會執行同步驗證，其中包括：

- 欄位層級檢查：我們驗證個別設定是否有正確的資料類型、允許值和格式需求。例如，我們確保時間值是正確的 Slurm 格式，而布林值使用接受的 Slurm 布林表示法。
- 內容感知驗證：針對更廣泛的組態內容檢查某些設定。例如，某些參數只有在啟用 Slurm 會計時才有效。
- 設定間一致性：我們確認互斥選項未一起設定，且已正確設定相互依存設定。

如果驗證失敗，您將收到ValidationException具有特定錯誤代碼（例如 InvalidInput）的、描述問題的明確錯誤訊息，以及無效欄位及其個別錯誤詳細資訊的清單。

雖然在此初始驗證期間發現許多問題，但只有在套用組態時，設定之間的一些複雜互動才會變得明顯。在這種情況下，操作會失敗並顯示資訊性錯誤訊息，而任何部分變更都會復原。

限制

AWS PCS 實作允許清單方法，以保護服務安全性和操作穩定性。會限制可能影響服務帳戶安全性或干擾受管服務功能的設定。不過，我們會持續評估客戶需求，並根據客戶意見回饋新增對其他設定的支援。

主題

- [AWS PCS 叢集的自訂 Slurm 設定](#)
- [AWS PCS 運算節點群組的自訂 Slurm 設定](#)
- [AWS PCS 佇列的自訂 Slurm 設定](#)
- [對 AWS PCS 中的自訂 Slurm 設定進行故障診斷](#)

AWS PCS 叢集的自訂 Slurm 設定

叢集層級支援下列自訂 Slurm 設定：

- [AccountingStorageEnforce](#)
- [AccountingStorageTRES](#)
- [AccountingStoreFlags](#)
- [DefMemPerCPU](#)
- [Epilog](#)
- [EnforcePartLimits](#)
- [FairShareDampeningFactor](#)
- [FirstJobId](#)
- [HealthCheckInterval](#)
- [HealthCheckNodeState](#)
- [HealthCheckProgram](#)
- [JobRequeue](#)
- [LaunchParameters](#)
- [Licenses](#)
- [MinJobAge](#)

Note

AWS PCS 支援的最小值為 5 秒MinJobAge。

- [OverTimeLimit](#)
- [PreemptExemptTime](#)
- [PreemptMode](#)

- [PreemptParameters](#)
- [PreemptType](#)
- [PriorityCalcPeriod](#)
- [PriorityDecayHalfLife](#)
- [PriorityFavorSmall](#)
- [PriorityFlags](#)
- [PriorityMaxAge](#)
- [PriorityUsageResetPeriod](#)
- [PriorityWeightAge](#)
- [PriorityWeightAssoc](#)
- [PriorityWeightFairshare](#)
- [PriorityWeightJobSize](#)
- [PriorityWeightPartition](#)
- [PriorityWeightQOS](#)
- [PriorityWeightTRES](#)
- [Prolog](#)
- [PrologFlags](#)
- [PropagatePrioProcess](#)
- [PropagateResourceLimits](#)
- [PropagateResourceLimitsExcept](#)
- [RequeueExit](#)
- [RequeueExitHold](#)
- [SchedulerParameters](#)
- [SelectTypeParameters](#)
- [SrunPortRange](#)
- [TaskEpilog](#)
- [TaskPluginParam](#)
- [TaskProlog](#)
- [TrackWCKey](#)
- [UnkillableStepProgram](#)

- [UnkillableStepTimeout](#)

AWS PCS 運算節點群組的自訂 Slurm 設定

運算節點群組層級支援下列自訂 Slurm 設定：

- [CpuSpecList](#)
- [Features](#)
- [MemSpecLimit](#)
- [RealMemory](#)
- [Weight](#)

AWS PCS 佇列的自訂 Slurm 設定

佇列層級支援下列自訂 Slurm 設定：

- [AllowAccounts](#)
- [AllowQoS](#)
- [Default](#)
- [DefaultTime](#)
- [DenyAccounts](#)
- [DenyQoS](#)
- [ExclusiveUser](#)
- [GraceTime](#)
- [MaxTime](#)
- [OverSubscribe](#)
- [OverTimeLimit](#)
- [PreemptMode](#)
- [PriorityJobFactor](#)
- [PriorityTier](#)
- [QOS](#)
- [TRESBillingWeights](#)

對 AWS PCS 中的自訂 Slurm 設定進行故障診斷

如果您在使用 Slurm 自訂設定建立或更新 AWS PCS 資源時發生錯誤，您可以使用記錄來診斷和解決問題。

對不相容的 Slurm 自訂設定進行故障診斷

問題：執行叢集、運算節點群組或佇列操作時，您會收到類似以下的錯誤訊息：

```
{OPERATION} failed. The Slurm custom settings of the cluster might be incompatible.  
Check the settings and try again.
```

下列操作可能發生此錯誤：

- CreateCluster
- CreateComputeNodeGroup
- UpdateComputeNodeGroup
- CreateQueue
- UpdateQueue

解決方案：啟用記錄以了解特定問題，並對不相容的設定進行故障診斷。

疑難排解不相容的 Slurm 自訂設定

1. 如果叢集尚不存在，請建立叢集，或確保現有叢集處於可啟用記錄的狀態。
2. 啟用叢集的記錄。如需詳細說明，請參閱 [AWS PCS 的記錄和監控](#)。

Note

建立叢集後，即可啟用記錄功能。

3. 檢閱日誌以識別導致不相容的特定 Slurm 組態問題。
4. 根據日誌資訊修正不相容的自訂設定，然後重試操作。

如需有關支援的 Slurm 自訂設定的資訊，請參閱：

- [AWS PCS 叢集的自訂 Slurm 設定](#)
- [AWS PCS 運算節點群組的自訂 Slurm 設定](#)

- [AWS PCS 佇列的自訂 Slurm 設定](#)

在 AWS PCS 中設定自訂 cgroup 設定

Slurm 使用 Linux cgroup 子系統來管理和限制任務的資源，包括記憶體、CPU 核心、裝置和交換空間。AWS PCS 可讓您在叢集建立或更新SlurmConfiguration期間，透過 CgroupCustomSettings 屬性在叢集層級自訂cgroup.conf設定。

設定 cgroup 設定

Cgroup 自訂設定可以在叢集建立期間透過 AWS 主控台、CLI 或 SDKs 進行設定，或在稍後透過更新操作進行修改。

AWS 管理主控台

導覽至叢集資源的建立或編輯頁面中的其他排程器設定。

新增設定

1. 選擇新增設定。
2. 從下拉式清單中選取參數名稱（其中包含簡短參數說明）。
3. 提供對應的值。

取消設定自訂設定

1. 選擇相關參數/值對旁的移除。
2. 建立或更新 資源。

AWS CLI

若要以程式設計方式管理 cgroup 設定，請在建立或更新叢集操作中使用 CgroupCustomSettings 欄位。

Example– 在叢集ConstrainRAMSpace上設定

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'CgroupCustomSettings=[{parameterName=ConstrainRAMSpace,parameterValue="yes"}]'
```

叢集支援的 cgroup 設定

叢集層級支援下列自訂 cgroup 設定：

- [AllowedRAMSpace](#)
- [AllowedSwapSpace](#)
- [ConstrainCores](#)
- [ConstrainDevices](#)
- [ConstrainRAMSpace](#)
- [ConstrainSwapSpace](#)
- [IgnoreSystemd](#)
- [MaxRAMPercent](#)
- [MaxSwapPercent](#)
- [MinRAMSpace](#)
- [SignalChildrenProcesses](#)

在 AWS PCS 中設定自訂 SlurmDBD 設定

Slurm 的資料庫協助程式 (slurmdbd) 管理會計資料、資料保留政策和隱私權控制。AWS PCS 可讓您在叢集建立或更新SlurmConfiguration期間，透過 SlurmdbdCustomSettings 屬性在叢集層級自訂slurmdbd.conf設定。

設定 slurmdbd 設定

Slurmdbd 自訂設定可以在叢集建立期間透過 AWS 主控台、CLI 或 SDKs 進行設定，或在稍後透過更新操作進行修改。

AWS 管理主控台

導覽至叢集資源的建立或編輯頁面中的其他排程器設定。

新增設定

1. 選擇新增設定。
2. 從下拉式清單中選取參數名稱（其中包含簡短參數說明）。
3. 提供對應的值。

取消設定自訂設定

1. 選擇相關參數/值對旁的移除。
2. 建立或更新 資源。

AWS CLI

若要以程式設計方式管理 slurmdbd 設定，請在建立或更新叢集操作中使用 SlurmdbdCustomSettings 欄位。

Example– 在叢集TrackWCKey上設定

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'SlurmdbdCustomSettings=[{parameterName=TrackWCKey,parameterValue="yes"}]'
```

叢集支援的 slurmdbd 設定

叢集層級支援下列自訂 slurmdbd 設定：

- [AllowNoDefAcct](#)
- [AllResourcesAbsolute](#)
- [CommitDelay](#)
- [DefaultQOS](#)
- [MaxQueryTimeRange](#)
- [Parameters](#)
- [PrivateData](#)
- [PurgeEventAfter](#)
- [PurgeJobAfter](#)
- [PurgeResvAfter](#)
- [PurgeStepAfter](#)
- [PurgeSuspendAfter](#)
- [PurgeTXNAfter](#)
- [PurgeUsageAfter](#)
- [TrackWCKey](#)

使用 SPANK 外掛程式擴展 AWS PCS 上的 Slurm 功能

使用 SPANK（適用於 Node 和 job Kontrol 的 Slurm 外掛程式架構）外掛程式，在 AWS PCS 叢集上的任務啟動和執行期間擴展和修改 Slurm 的行為。SPANK 外掛程式提供一般界面來攔截和修改任務啟動階段。

在運算節點 AMI 上安裝 SPANK 外掛程式，並設定它們來根據您的工作負載需求自訂 Slurm 叢集的行為。如需 SPANK 的詳細資訊，請參閱 SchedMD 網站上的 [SPANK 文件](#)。

內容

- [在 AWS PCS 上安裝 SPANK 外掛程式](#)
- [在 AWS PCS 上設定 SPANK 外掛程式](#)
- [AWS PCS 上 SPANK 外掛程式的常見問題](#)

在 AWS PCS 上安裝 SPANK 外掛程式

遵循外掛程式的文件，在您的 AMI 上安裝 SPANK 外掛程式。

在叢集上編譯特定 Slurm 版本的 SPANK 外掛程式。AWS PCS 提供的 Slurm 安裝程式會將 Slurm 存放在 `中/opt/aws/pcs/scheduler/slurm-version`。當您編譯外掛程式時，請指定 Slurm 版本。

下列範例顯示如何指定某些外掛程式的 Slurm 版本：

```
export CFLAGS="-I/opt/aws/pcs/scheduler/slurm-version/include"
```

如果您在 AMI 中有多個 Slurm 版本，請編譯每個版本的外掛程式。將編譯的外掛程式存放在版本控制的資料夾中。

下列範例顯示如何指定某些外掛程式的目的地資料夾：

```
export DESTDIR="your-preferred-versioned-path"
```

Important

外掛程式可能需要不同的變數。請參閱您要安裝之外掛程式的官方文件。

在 AWS PCS 上設定 SPANK 外掛程式

根據預設，將組態檔案存放在 `/etc/aws/pcs/scheduler/slurm-version/plugstack.conf.d/`。

若要將您的 SPANK 組態儲存在不同的位置，請將您的位置新增至預設目錄中的組態檔案。

下列範例示範如何包含來自其他目錄的組態檔案：

```
# content of /etc/aws/pcs/scheduler/slurm-version/any-filename.conf
include path-to-your-configuration-folder/*.conf
include path-to-a-second-configuration-folder/*.conf
```

將每個組態儲存在專用檔案中，或常見檔案中。您可以使用多個組態檔案。

下列範例顯示範例組態檔案：

```
# content of path-to-your-or-default-config-folder/filename-1.conf
required path-to-plugin-1 arguments
optional path-to-plugin-2 arguments
```

```
# content of path-to-your-or-default-config-folder/filename-2.conf
required path-to-plugin-3 arguments
```

如需如何設定外掛程式的詳細資訊，請參閱 SchedMD 網站上的 [SPANK 組態文件](#)。

Important

設定資料夾許可，以防止未經授權的外掛程式組態變更。

Note

AWS PCS 不會管理您的 SPANK 外掛程式。如果您收到與外掛程式相關的錯誤，請檢查運算節點上的錯誤日誌。

Note

當載入 SPANK 組態時，Slurm 錯誤地記錄類似下列的錯誤：

```
error: "Include" failed in file /etc/slurm/plugstack.conf line 3
```

您可以忽略此錯誤。這不會影響 SPANK 外掛程式的運作方式。

AWS PCS 上 SPANK 外掛程式的常見問題

本節說明在 AWS PCS 叢集上安裝和設定 SPANK 外掛程式的常見問題。

我需要在登入節點和運算節點上安裝 SPANK 外掛程式嗎？

有些 SPANK 外掛程式不需要在所有節點上安裝；但為了提高相容性，建議您在每個節點上安裝所有 SPANK 外掛程式。

生產使用 SPANK 外掛程式需要哪些額外的組態？

除了範例中顯示的基本安裝和組態之外，生產部署通常需要額外的設定。容器型外掛程式，例如，Pyxis 可能會要求您設定 Enroot 的環境變數、啟用 PMI（程序管理界面），以及設定容器執行時間的許可。如需詳細的生產部署需求，請參閱特定外掛程式的文件。

如何疑難排解 SPANK 外掛程式問題？

AWS PCS 不會管理 SPANK 外掛程式。檢查運算節點上的錯誤日誌，以疑難排解問題。

使用 Slurm CLI 篩選條件外掛程式在 AWS PCS 中自訂任務提交

AWS PCS 支援 Slurm CLI Filter Plugins 執行自訂 Lua 指令碼，以驗證和修改登入和運算節點上的任務提交參數。如需 CLI 篩選條件外掛程式的詳細資訊，請參閱 SchedMD 網站上的 [cli_filter 外掛程式 API 文件](#)。

要求

CLI 篩選條件外掛程式需要 Slurm 24.11 版或更新版本，以及部署至所有登入和運算節點的 Lua 指令碼。

Important

對於 Slurm 24.11 和 25.05 版，CLI 篩選條件外掛程式需要使用 AWS PCS Slurm 安裝程式 (24.11.6-2+ 或 25.05.4-1+ 版) 安裝 Slurm。如需安裝 Slurm 的詳細資訊，請參閱 [步驟 3 – 安裝 Slurm](#)。

限制和安全考量

- 安全強制執行 – 任何使用者都可以輕鬆略過 CLI 篩選條件外掛程式，且不得用於安全關鍵政策。使用者可以透過提供在提交任務時停用的自訂組態來 `CLIFilterPlugins` 停用 CLI 篩選外掛程式。
- 僅限 Lua 實作 – 支援 Lua 指令碼實作。不支援 C 實作。

主題

- [在 AWS PCS 叢集上設定 Slurm CLI 篩選條件外掛程式](#)
- [使用 Amazon S3 在 AWS PCS 中部署 CLI 篩選條件外掛程式指令碼](#)
- [翻譯 Slurm 任務提交外掛程式指令碼，以在 AWS PCS 中使用 CLI 篩選條件外掛程式](#)
- [有關 AWS PCS 中 Slurm CLI 篩選條件外掛程式的常見問題](#)
- [疑難排解 AWS PCS 中的 Slurm CLI 篩選條件外掛程式問題](#)

在 AWS PCS 叢集上設定 Slurm CLI 篩選條件外掛程式

在建立新的 AWS PCS 叢集時設定 CLI 篩選條件外掛程式。您可以使用更新 API 或主控台在現有叢集上啟用或停用 CLI 篩選條件外掛程式，而無需重新建立叢集。

先決條件

設定 CLI 篩選條件外掛程式之前，請先完成下列任務：

- 撰寫並測試實作 CLI Filter Plugin API 的 Lua 指令碼
- 準確命名 Lua 指令碼 `cli_filter.lua`
- 選擇將指令碼部署到所有叢集執行個體 (AMI、S3 或檔案系統) 的方法
- 確認您使用的是 Slurm 24.11 版或更新版本

在新叢集上啟用 CLI 篩選條件外掛程式

AWS PCS console

1. 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
2. 在導覽窗格中，選擇叢集。
3. 選擇 Create Cluster (建立叢集)。
4. 選取有效的 Slurm 版本 (24.11 版或更新版本)。
5. 在排程器設定下，展開其他排程器設定。
6. 新增參數名稱設為 CliFilterPlugins 且參數值設為 的新 Slurm 自訂設定 cli_filter/ lua。
7. 完成剩餘的叢集組態，然後選擇建立叢集。

AWS PCS API

在您呼叫 CreateCluster API 動作時提供 slurmCustomSettings 組態。將 parameterName 設定為 CliFilterPlugins，將 parameterValue 設定為 cli_filter/ lua。如需詳細資訊，請參閱 AWS PCS API 參考中的 [CreateCluster](#)。

下列範例使用 AWS CLI 呼叫 CreateCluster API 動作。自訂設定 CliFilterPlugins=cli_filter/ lua 會啟用 CLI 篩選條件外掛程式。

```
aws pcs create-cluster --cluster-name cluster-name \  
--scheduler type=SLURM,version=24.11 \  
--size SMALL \  
--networking subnetIds=cluster-subnet-id,securityGroupIds=cluster-security-group-id \  
\   
--slurm-configuration \  
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/  
lua"}]'
```

部署 CLI 篩選條件外掛程式指令碼

將 CLI 篩選條件外掛程式指令碼部署到您的叢集

1. 確保運算節點群組中使用的所有 AMIs 透過 AWS PCS Slurm 安裝程式安裝 Slurm。

Note

如果您對所有運算節點群組使用 AWS PCS 範例 AMI，請略過此步驟。Slurm 已安裝。

- 將您的 `cli_filter.lua` 指令碼部署 `/etc/aws/pcs/scheduler/slurm-<version>/cli_filter.lua` 到叢集中所有執行個體上的。

例如，對於 Slurm 24.11 版：

```
/etc/aws/pcs/scheduler/slurm-24.11/cli_filter.lua
```

- 使用您準備好 AMIs 啟動所有登入和運算節點。
- 測試任務提交以驗證 CLI 篩選條件外掛程式是否正確執行。

在現有叢集上啟用或停用 CLI 篩選條件外掛程式

您可以在現有叢集上啟用或停用 CLI 篩選條件外掛程式，而無需重建您的基礎設施。如需詳細資訊，請參閱 [在 AWS PCS 中更新叢集](#)。

AWS PCS console

- 在 <https://console.aws.amazon.com/pcs/> 開啟 AWS PCS 主控台。
- 在導覽窗格中，選擇叢集。
- 選取要更新的叢集。
- 選擇編輯動作。
- 在編輯叢集頁面的其他排程器設定下：
 - 若要啟用 CLI 篩選條件外掛程式：新增參數名稱設定為 `CliFilterPlugins` 且參數值設定為的新 Slurm 自訂設定 `cli_filter/lua`。
 - 若要停用 CLI 篩選條件外掛程式：移除現有 `CliFilterPlugins` 設定。
- 選擇更新叢集以提交變更。
- 監控叢集狀態，其會在程序期間顯示為「更新」，並在更新完成時顯示為「作用中」。

AWS PCS API

使用 UpdateCluster API 動作啟用或停用 CLI 篩選條件外掛程式。如需詳細資訊，請參閱 AWS PCS API 參考中的 [UpdateCluster](#)。

若要在現有叢集上啟用 CLI 篩選條件外掛程式：

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/  
lua"}]'
```

若要在現有叢集上停用 CLI 篩選條件外掛程式：

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'slurmCustomSettings=[]'
```

預期的結果

完成組態之後：

- 在開啟 CLI Filter Plugin 的情況下建立叢集
- 任務提交會在到達 Slurm 控制器之前觸發您的自訂驗證邏輯
- 不合規任務會因您的自訂錯誤訊息而遭到拒絕
- 合規任務透過 Slurm 排程器正常進行

疑難排解

任何節點上缺少 CLI 篩選條件外掛程式指令碼

症狀：任務提交會立即失敗，並出現外掛程式載入錯誤。

可能原因：指令碼未部署到所有執行個體或不正確的檔案路徑或名稱。

解決方法：確認指令碼存在於所有登入和運算節點的正确路徑，且檔案名稱為 `cli_filter.lua`。

無效的 CLI 篩選條件外掛程式組態

徵狀：叢集建立失敗並出現驗證錯誤。

可能原因：CliFilterPlugins 參數未設定為 cli_filter/lua 格式。

解決方案：在 cli_filter/lua 中使用確切的參數值 slurmCustomSettings。

使用 Amazon S3 在 AWS PCS 中部署 CLI 篩選條件外掛程式指令碼

當您想要在即時叢集上更新任務提交邏輯而不重建 AMIs 時，請使用 S3 部署 CLI 篩選條件外掛程式指令碼。此方法會使用使用者資料，在執行個體啟動期間從 S3 下載指令碼。

先決條件

使用 S3 部署指令碼之前，請先完成下列任務：

- 使用 CLI Filter Plugin Lua 指令碼建立 S3 儲存貯體
- 設定具有 S3 儲存貯體讀取存取權的 IAM 執行個體描述檔
- 設定 S3 VPC Gateway 端點以不使用網際網路直接存取
- 準備要從 S3 下載的使用者資料指令碼

使用 S3 部署 CLI 篩選條件外掛程式指令碼

1. 將 cli_filter.lua 指令碼上傳至 S3 儲存貯體。
2. 使用儲存貯體的 S3 讀取許可來設定 IAM 執行個體描述檔。
3. 將 shell 程式碼新增至啟動範本使用者資料，以下載指令碼：

```
aws s3 cp s3://my-bucket/cli_filter.lua /etc/aws/pcs/scheduler/slurm-24.11/
cli_filter.lua
chmod 644 /etc/aws/pcs/scheduler/slurm-24.11/cli_filter.lua
```

4. 使用更新的啟動範本部署運算節點群組。
5. 測試任務提交以驗證指令碼功能。

預期的結果

完成 S3 部署之後：

- CLI Filter Plugin 指令碼會在啟動期間自動下載到所有執行個體
- S3 中的指令碼更新反映在新啟動的執行個體上
- 在整個叢集中一致地強制執行任務提交政策

疑難排解

S3 存取遭拒

徵狀：執行個體啟動失敗或未下載指令碼。

可能原因：缺少 IAM 許可或 S3 VPC 端點。

解決方案：確認 IAM 執行個體描述檔具有 `s3:GetObject` 許可，並已設定 S3 VPC 端點。

翻譯 Slurm 任務提交外掛程式指令碼，以在 AWS PCS 中使用 CLI 篩選條件外掛程式

當您從其他 Slurm 環境遷移時，將現有的任務提交外掛程式 Lua 指令碼轉換為 CLI 篩選條件外掛程式。轉譯程序涉及更新函數名稱和欄位存取模式，以使用 CLI Filter Plugin API。

先決條件

翻譯指令碼之前，請先完成下列任務：

- 檢閱您現有的任務提交外掛程式 Lua 指令碼
- 了解任務提交和 CLI 篩選條件外掛程式 APIs 之間的差異
- 存取 Slurm CLI 篩選條件外掛程式文件

將任務提交外掛程式指令碼轉譯為 CLI 篩選條件外掛程式

1. 檢閱您現有的任務提交外掛程式指令碼函數 (`slurm_job_submit`、`slurm_job_modify`)。
2. 識別對等的 CLI 篩選條件外掛程式函數：
 - `slurm_job_submit` 成為 `slurm_cli_pre_submit`
 - 新增 `slurm_cli_setup_defaults` 預設參數設定
 - 針對提交 `slurm_cli_post_submit` 後動作新增

3. 將任務驗證邏輯從 `job_desc` 欄位轉換為 `options` 陣列存取：
 - `job_desc.account` 成為 `options["account"]`
 - `job_desc.partition` 成為 `options["partition"]`
 - `job_desc.features` 成為 `options["constraint"]`
4. 將日誌記錄呼叫從 更新 `slurm.log_user()` 為 `slurm.log_error()`。
5. 在開發叢集上測試您的翻譯指令碼。
6. 遵循標準 CLI 篩選條件外掛程式部署程序，部署到您的生產叢集。

預期的結果

完成翻譯之後：

- 您的翻譯指令碼提供對等任務提交驗證
- 使用者會看到與原始任務提交外掛程式類似的錯誤訊息和提示
- 任務提交政策會在遷移至 AWS PCS 期間維護

疑難排解

指令碼轉譯錯誤

症狀：任務提交失敗並出現 Lua 執行錯誤。

可能原因：翻譯指令碼中的欄位存取或函數呼叫不正確。

解決方案：檢閱 CLI 篩選條件外掛程式 API 文件，並比較任務提交和 CLI 篩選條件介面之間的欄位映射。

有關 AWS PCS 中 Slurm CLI 篩選條件外掛程式的常見問題

檢閱這些有關 CLI 篩選條件外掛程式的常見問題。

CLI 篩選條件外掛程式和任務提交外掛程式之間的差異是什麼？

CLI Filter Plugin 會在登入時執行用戶端，並在任務提交到達控制器之前運算節點，而 Job Submit Plugin 會在任務提交後於控制器上執行伺服器端。使用者可以略過 CLI 篩選條件外掛程式，但不保留控制器鎖定，而任務提交是安全的，但可能會影響執行期間的叢集效能。

AWS PCS 是否支援 Slurm 任務提交外掛程式？

否，AWS PCS 不支援任務提交外掛程式。使用 CLI 篩選條件外掛程式來驗證和修改任務提交。

我可以將 CLI 篩選條件外掛程式強制執行安全性嗎？

否，CLI 篩選條件外掛程式可由確定的使用者略過，不應依賴進行安全強制執行。將其用於使用者體驗改進、預設參數設定和政策指導，而不是安全關鍵政策。

為什麼指令碼必須位於所有運算節點上，而不只是登入節點？

這類 Slurm 命令 `srun` 可以在運算節點的任務指令碼中執行，這也觸發 CLI Filter Plugin 執行。指令碼必須可在執行 Slurm 命令的位置使用。

我可以修改即時叢集上的 CLI 篩選條件外掛程式指令碼嗎？

是，如果您使用 S3 或檔案系統部署方法。新的執行個體將取得更新的指令碼，但現有的執行個體需要手動或透過您選擇的部署方法更新指令碼。

我可以在不同運算節點群組上使用不同的 CLI 篩選條件外掛程式指令碼嗎？

是，但不建議這麼做。您可以向不同的運算節點群組提供具有不同邏輯的指令碼，但您負責管理相互依存性並防止邏輯重疊。大多數客戶在整個叢集中提供一組邏輯。

我可以將 CLI 篩選條件外掛程式與 C 實作搭配使用，而不是使用 Lua 嗎？

不支援 C 實作。AWS PCS 僅支援 Lua 指令碼實作。SchedMD 建議客戶透過 C 使用 Lua，以便在實作 CLI 篩選條件外掛程式時使用。

我可以在現有叢集上開啟或關閉 CLI 篩選條件外掛程式嗎？

可以，您可以使用更新 API 在現有叢集上啟用或停用 CLI 篩選條件外掛程式，而無需重新建立叢集。

疑難排解 AWS PCS 中的 Slurm CLI 篩選條件外掛程式問題

使用此故障診斷資訊來解決常見的 CLI 篩選條件外掛程式問題。

任務提交立即失敗，並出現外掛程式載入錯誤

徵狀：使用者在提交任務時，會收到有關缺少或失敗 CLI 篩選條件外掛程式的錯誤訊息。

可能原因：

- 一或多個節點缺少 CLI 篩選條件外掛程式指令碼

- 不正確的指令碼檔案名稱 (必須剛好為 `cli_filter.lua`)
- 部署到錯誤目錄路徑的指令碼
- 指令碼具有不正確的檔案許可

解決方法：

- 確認指令碼存在於所有登入和運算節點/`etc/aws/pcs/scheduler/slurm-<version>/cli_filter.lua`上的
- 檢查指令碼檔案名稱是否完全相同 `cli_filter.lua`
- 確保指令碼具有可讀取的許可 (644 或類似許可)
- 在單一登入節點上測試指令碼部署，然後再部署到完整叢集

叢集建立失敗，並出現 CLI 篩選條件外掛程式驗證錯誤

徵狀：叢集建立失敗，並出現 `CliFilterPlugins` 參數無效的錯誤。

可能原因：

- 中的參數值格式不正確 `slurmCustomSettings`
- 參數名稱或值中的 Typo

解決方法：

- 使用確切的參數名稱：`CliFilterPlugins`
- 使用確切參數值：`cli_filter/lua`
- 在 `slurmCustomSettings` 陣列中驗證 JSON 語法

CLI 篩選條件外掛程式指令碼會執行，但任務驗證無法如預期般運作

症狀：任務提交成功，但自訂驗證邏輯不會觸發或產生非預期的結果。

可能原因：

- Lua 指令碼語法錯誤
- 欄位存取模式不正確 (使用任務提交外掛程式語法而非 CLI 篩選條件外掛程式)
- 驗證條件中的邏輯錯誤

解決方法：

- 檢閱 Lua 指令碼是否有語法錯誤
- 驗證欄位存取使用 `options["field_name"]` 格式，而不是 `job_desc.field_name`
- 新增記錄陳述式以偵錯指令碼執行流程

- 先使用簡單的驗證案例測試指令碼邏輯

S3 指令碼部署失敗

徵狀：執行個體啟動，但未從 S3 下載 CLI 篩選條件外掛程式指令碼。

可能原因：

- IAM 執行個體描述檔缺少 S3 讀取許可
- 未設定 S3 VPC 端點
- 使用者資料中的 S3 儲存貯體或物件路徑不正確

解決方法：

- 確認 IAM 執行個體描述檔具有儲存貯體的 `s3:GetObject` 許可
- 設定 S3 VPC Gateway 端點以進行直接存取
- 檢查使用者資料指令碼中的 S3 儲存貯體名稱和物件路徑
- 檢閱執行個體使用者資料日誌是否有 S3 下載錯誤

AWS 平行運算服務的安全性

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

安全性是 AWS 與您之間共同責任。[共同責任模式](#)將其描述為雲端的安全性，和雲端中的安全性：

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

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

主題

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

AWS 平行運算服務中的資料保護

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

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

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

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

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

靜態加密

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

叢集秘密存放在 AWS Secrets Manager，並使用 Secrets Manager 受管 KMS 金鑰加密。如需詳細資訊，請參閱[在 AWS PCS 中使用叢集秘密](#)。

在 AWS PCS 叢集中，下列資料為靜態：

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

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

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

傳輸中加密

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

AWS PCS 使用 TLS 連線到其他 AWS 服務。

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

金鑰管理

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

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

網際網路流量隱私權

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

您也可以使用下列方案，將內部部署網路連線至 AWS：

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

- AWS Direct Connect。如需詳細資訊，請參閱[什麼是 AWS Direct Connect？](#) AWS Direct Connect 《使用者指南》中的。

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

加密 API 流量

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

加密資料流量

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

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

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

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

Note

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

目錄

- [概要](#)
- [設定金鑰政策](#)
- [範例 1：允許存取客戶受管金鑰的金鑰政策區段](#)
- [範例 2：允許跨帳戶存取客戶受管金鑰的金鑰政策區段](#)

- [在 AWS KMS 主控台中編輯金鑰政策](#)

概要

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

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

Note

金鑰必須為對稱金鑰。Amazon EBS 不支援非對稱客戶受管金鑰。

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

設定金鑰政策

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

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

金鑰政策必須具有下列陳述式：

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

當您將新的政策陳述式新增至金鑰政策時，請勿變更政策中的任何現有陳述式。

如需詳細資訊，請參閱：

- AWS CLI 命令參考中的 [create-key](#)

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

範例 1：允許存取客戶受管金鑰的金鑰政策區段

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

```
{
  "Sid": "Allow service-linked role use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleForPCS"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleForPCS"
    ]
  },
}
```

```

"Action": [
  "kms:CreateGrant"
],
"Resource": "*",
"Condition": {
  "Bool": {
    "kms:GrantIsForAWSResource": true
  }
}
}

```

範例 2：允許跨帳戶存取客戶受管金鑰的金鑰政策區段

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

授予金鑰的存取權

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

```

{.
  "Sid": "Allow external account 111122223333 use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:root"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}

```

```
{
```

```

    "Sid": "Allow attachment of persistent resources in external
account 111122223333",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:root"
      ]
    },
    "Action": [
      "kms:CreateGrant"
    ],
    "Resource": "*"
  }
}

```

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

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

```

aws kms create-grant \
  --region us-west-2 \
  --key-id arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
  --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
pcs.amazonaws.com/AWSServiceRoleForPCS \
  --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
"GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"

```

Note

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

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

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowCreationOfGrantForTheKMSKeyInExternalAccount444455556666",
    "Effect": "Allow",
    "Action": "kms:CreateGrant",
    "Resource": "arn:aws:kms:us-west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
  }
]
```

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

Important

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

在 AWS KMS 主控台中編輯金鑰政策

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

Warning

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

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

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

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

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的 [AWS 服務 透過 存取 AWS PrivateLink](#)。

的考量事項 AWS PCS

在您設定的介面端點之前 AWS PCS，請檢閱《AWS PrivateLink 指南》中的 [使用介面 VPC 端點存取 AWS 服務](#)。

AWS PCS 支援透過介面端點呼叫其所有 API 動作。

如果您的 VPC 沒有直接網際網路存取，您必須設定 VPC 端點，讓運算節點群組執行個體呼叫 [RegisterComputeNodeGroupInstance](#) API AWS PCS 動作。

建立的介面端點 AWS PCS

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

AWS PCS 使用下列服務名稱建立的介面端點：

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

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

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

為您的介面端點建立端點政策

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

端點政策會指定以下資訊：

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

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的「[使用端點政策控制對服務的存取](#)」。

範例：AWS PCS 動作的 VPC 端點政策

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

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

AWS 平行運算服務的 Identity and Access Management

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

主題

- [目標對象](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [AWS Parallel Computing Service 如何與 IAM 搭配使用](#)
- [AWS Parallel Computing Service 的身分型政策範例](#)
- [AWS AWS Parallel Computing Service 的 受管政策](#)
- [AWS PCS 的服務連結角色](#)
- [AWS PCS 的 Amazon EC2 Spot 角色](#)
- [AWS PCS 的最低許可](#)
- [平行運算服務的 IAM AWS 執行個體描述檔](#)
- [平行 AWS 運算服務身分和存取的故障診斷](#)

目標對象

使用方式 AWS Identity and Access Management (IAM) 會根據您的角色而有所不同：

- 服務使用者 — 若無法存取某些功能，請向管理員申請所需許可 (請參閱 [平行 AWS 運算服務身分和存取的故障診斷](#))
- 服務管理員 — 負責設定使用者存取權並提交相關許可請求 (請參閱 [AWS Parallel Computing Service 如何與 IAM 搭配使用](#))
- IAM 管理員 — 撰寫政策以管理存取控制 (請參閱 [AWS Parallel Computing Service 的身分型政策範例](#))

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的[如何登入您的 AWS 帳戶](#)。

對於程式設計存取，AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的[API 請求的AWS 第 4 版簽署程序](#)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分具有對所有 AWS 服務和資源的完整存取權。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的[需要根使用者憑證的任務](#)。

聯合身分

最佳實務是要求人類使用者使用聯合身分提供者，以 AWS 服務使用臨時憑證存取。

聯合身分是您企業目錄、Web 身分提供者的使用者，或使用來自身分來源的 AWS 服務憑證存取 Directory Service。聯合身分會擔任角色，而該角色會提供臨時憑證。

若需集中化管理存取權限，建議使用 AWS IAM Identity Center。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的[什麼是 IAM Identity Center?](#)。

IAM 使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的[要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取](#)。

[IAM 群組](#)會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 使用者的使用案例](#)。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過[從使用者切換到 IAM 角色（主控台）](#)或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[擔任角色的方法](#)。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 中的快帳戶資源存取](#)。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS，會評估這些政策。大多數政策會以 JSON 文件 AWS 形式存放在中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的[JSON 政策概觀](#)。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的[透過客戶管理政策定義自訂 IAM 許可](#)。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的[在受管政策與內嵌政策之間選擇](#)。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。您必須在資源型政策中[指定主體](#)。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 實體許可界限](#)。
- 服務控制政策 (SCP) — 為 AWS Organizations 中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[服務控制政策](#)。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[資源控制政策 \(RCP\)](#)。
- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參閱《IAM 使用者指南》中的[工作階段政策](#)。

多種政策類型

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

AWS Parallel Computing Service 如何與 IAM 搭配使用

在您使用 IAM 管理 AWS PCS 的存取權之前，請先了解哪些 IAM 功能可與 AWS PCS 搭配使用。

您可以搭配 AWS 平行運算服務使用的 IAM 功能

IAM 功能	AWS PCS 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
主體許可	是
服務角色	否
服務連結角色	是

若要全面了解 AWS PCS 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《IAM 使用者指南》中的與 IAM [AWS 搭配使用的服務](#)。

AWS PCS 的身分型政策

支援身分型政策：是

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

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素參考](#)。

AWS PCS 的身分型政策範例

若要檢視 AWS PCS 身分型政策的範例，請參閱 [AWS Parallel Computing Service 的身分型政策範例](#)。

AWS PCS 內的資源型政策

支援資源型政策：否

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

如需啟用跨帳戶存取權，您可以在其他帳戶內指定所有帳戶或 IAM 實體作為資源型政策的主體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的快帳戶資源存取](#)。

AWS PCS 的政策動作

支援政策動作：是

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

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS PCS 動作清單，請參閱《服務授權參考》中的 [AWS 平行運算服務定義的動作](#)。

AWS PCS 中的政策動作在動作之前使用以下字首：

```
pcs
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
```

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

AWS PCS 的政策資源

支援政策資源：是

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

Resource JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

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

若要查看 AWS PCS 資源類型及其 ARNs，請參閱《服務授權參考》中的[AWS 平行運算服務定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱[AWS 平行運算服務定義的動作](#)。

若要檢視 AWS PCS 身分型政策的範例，請參閱 [AWS Parallel Computing Service 的身分型政策範例](#)。

AWS PCS 的政策條件索引鍵

支援服務特定政策條件金鑰：是

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

Condition 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的[AWS 全域條件內容索引鍵](#)。

若要查看 AWS PCS 條件金鑰清單，請參閱《服務授權參考》中的[AWS 平行運算服務的條件金鑰](#)。若要了解您可以使用條件索引鍵的動作和資源，請參閱[AWS 平行運算服務定義的動作](#)。

若要檢視 AWS PCS 身分型政策的範例，請參閱 [AWS Parallel Computing Service 的身分型政策範例](#)。

AWS PCS ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 搭配 AWS PCS

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，依據稱為標籤的屬性來定義許可。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在委託人的標籤符合資源上的標籤時允許操作。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的 [使用 ABAC 授權定義許可](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的 [使用屬性型存取控制 \(ABAC\)](#)。

搭配 AWS PCS 使用臨時登入資料

支援臨時憑證：是

臨時登入資料提供對 AWS 資源的短期存取，並在您使用聯合或切換角色時自動建立。AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的臨時安全憑證與可與 IAM 搭配運作的 AWS 服務](#)。

AWS PCS 的跨服務主體許可

支援轉寄存取工作階段 (FAS)：是

轉送存取工作階段 (FAS) 使用呼叫的委託人許可 AWS 服務，結合 AWS 服務請求向下游服務提出請求。如需提出 FAS 請求時的政策詳細資訊，請參閱 [轉發存取工作階段](#)。

AWS PCS 的服務角色

支援服務角色：否

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

Warning

變更服務角色的許可可能會中斷 AWS PCS 功能。只有在 AWS PCS 提供指引時，才能編輯服務角色。

AWS PCS 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 AWS 帳戶，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理 AWS PCS 服務連結角色的詳細資訊，請參閱 [AWS PCS 的服務連結角色](#)。

AWS Parallel Computing Service 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 AWS PCS 資源的許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策 \(主控台\)](#)。

如需 AWS PCS 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的[AWS 平行運算服務的動作、資源和條件金鑰](#)。

主題

- [政策最佳實務](#)
- [使用 AWS PCS 主控台](#)
- [允許使用者檢視他們自己的許可](#)

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AWS PCS 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [使用 IAM Access Analyzer 驗證政策](#)。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [透過 MFA 的安全 API 存取](#)。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 [IAM 安全最佳實務](#)。

使用 AWS PCS 主控台

若要存取 AWS 平行運算服務主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AWS PCS 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

如需使用 AWS PCS 主控台所需最低許可的詳細資訊，請參閱 [AWS PCS 的最低許可](#)。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ViewOwnUserInfo",
    "Effect": "Allow",
    "Action": [
      "iam:GetUserPolicy",
      "iam:ListGroupsForUser",
      "iam:ListAttachedUserPolicies",
      "iam:ListUserPolicies",
      "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
  },
  {
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
      "iam:GetGroupPolicy",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedGroupPolicies",
      "iam:ListGroupPolicies",
      "iam:ListPolicyVersions",
      "iam:ListPolicies",
      "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
```

AWS AWS Parallel Computing Service 的 受管政策

AWS 受管政策是由 AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住，AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義特定於使用案例的[客戶管理政策](#)，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。當新的 AWS 服務 啟動或新的 API 操作可供現有服務使用時，AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#)。

AWS 受管政策：AWSPCSComputeNodePolicy

您可以將 AWSPCSComputeNodePolicy 連接至 IAM 實體。您可以將此政策連接至您指定的 AWS PCS 運算節點 IAM 角色，以允許使用該角色連線至 AWS PCS 叢集的節點。

AWS 當您使用主控台建立運算節點群組時，PCS 會將此政策連接至運算節點群組角色。

許可詳細資訊

此政策包含以下許可。

- `pcs:RegisterComputeNodeGroupInstance` – 允許 AWS PCS 運算節點 (EC2 執行個體) 向 AWS PCS 叢集註冊。

若要檢視此政策的許可，請參閱《AWS 受管政策參考》中的 [AWSPCSComputeNodePolicy](#)。

AWS 受管政策：AWSPCSServiceRolePolicy

您無法將 AWSPCSServiceRolePolicy 連接至 IAM 實體。此政策會連接到服務連結角色，允許 AWS PCS 代表您執行動作。如需詳細資訊，請參閱[AWS PCS 的服務連結角色](#)。

許可詳細資訊

此政策包含以下許可。

- `ec2` – 允許 AWS PCS 建立和管理 Amazon EC2 資源。

- iam – 允許 AWS PCS 為 Amazon EC2 機群建立服務連結角色，並將角色傳遞給 Amazon EC2。
- cloudwatch – 允許 AWS PCS 將服務指標發佈至 Amazon CloudWatch。
- secretsmanager – 允許 AWS PCS 管理 AWS PCS 叢集資源的秘密。

若要檢視此政策的許可，請參閱《AWS 受管政策參考》中的 [AWSPCSServiceRolePolicy](#)。

AWS AWS 受管政策的 PCS 更新

檢視自此服務開始追蹤這些變更以來 AWS ，PCS AWS 受管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 AWS PCS 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	Date
AWSPCSServiceRolePolicy – 更新至現有政策	<p>AWS PCS 新增了新的許可，以支援可預測運算容量的容量區塊。</p> <p>新增ec2:DescribeCapacityReservations 許可，讓 AWS PCS 能夠探索和使用運算節點群組的容量區塊保留。</p>	2025 年 9 月 11 日
AWSPCSComputeNodePolicy – 新政策	<p>AWS PCS 新增了新的政策，以授予 AWS PCS 運算節點連線至 AWS PCS 叢集的許可。</p> <p>AWS 當您在 PCS 主控台中建立運算節點群組時，AWS PCS 會將此政策連接至 IAM 角色。</p>	2025 年 6 月 23 日
更新本文件中的 JSON	更正本文件中的 JSON 以包含 "arn:aws:ec2:*:*:spot-instances-request/*" 。	2024 年 9 月 5 日
AWS PCS 已開始追蹤變更	AWS PCS 開始追蹤其 AWS 受管政策的變更。	2024 年 8 月 28 日

AWS PCS 的服務連結角色

AWS 平行運算服務使用 AWS Identity and Access Management (IAM) [服務連結角色](#)。服務連結角色是直接連結至 AWS PCS 的唯一 IAM 角色類型。服務連結角色由 AWS PCS 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定 AWS PCS，因為您不必手動新增必要的許可。AWS PCS 會定義其服務連結角色的許可，除非另有定義，否則只有 AWS PCS 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除角色的相關資源，才能刪除服務連結角色。這可保護您的 AWS PCS 資源，因為您不會意外移除存取資源的許可。

如需有關支援服務連結角色的其他服務的資訊，請參閱[AWS 使用 IAM 的服務](#)，並在服務連結角色欄中尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

AWS PCS 的服務連結角色許可

AWS PCS 使用名為 `AWSServiceRoleForPCS` 的服務連結角色 – 授予 AWS PCS 管理 Amazon EC2 資源的許可。

`AWSServiceRoleForPCS` 服務連結角色信任下列服務擔任該角色：

- `pcs.amazonaws.com`

名為 [AWSPCSServiceRolePolicy](#) 的角色許可政策允許 AWS PCS 對特定資源完成動作。

您必須設定許可，以允許您的使用者、群組或角色建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的[服務連結角色許可](#)。

為 AWS PCS 建立服務連結角色

您不需要手動建立服務連結角色。AWS PCS 會在您建立叢集時為您建立服務連結角色。

編輯 AWS PCS 的服務連結角色

AWS PCS 不允許您編輯 `AWSServiceRoleForPCS` 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的[編輯服務連結角色](#)。

刪除 AWS PCS 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

Note

如果 AWS PCS 服務在您嘗試刪除資源時使用角色，則刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

移除 AWSServiceRoleForPCS 使用的 AWS PCS 資源 AWSServiceRoleForPCS

您必須刪除所有叢集，才能刪除 AWSServiceRoleForPCS 服務連結角色。如需詳細資訊，請參閱[刪除叢集](#)。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台 AWS CLI、或 AWS API 來刪除 AWSServiceRoleForPCS 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[刪除服務連結角色](#)。

AWS PCS 服務連結角色支援的 區域

AWS PCS 支援在所有提供服務的區域中使用服務連結角色。如需詳細資訊，請參閱[AWS Regions and endpoints](#)。

AWS PCS 的 Amazon EC2 Spot 角色

如果您想要建立使用 Spot 作為其購買選項的 AWS PCS 運算節點群組，您還必須在 中擁有 AWSServiceRoleForEC2Spot 服務連結角色 AWS 帳戶。您可以使用下列 AWS CLI 命令來建立角色。如需詳細資訊，請參閱AWS Identity and Access Management 《使用者指南》中的[建立服務連結角色](#)和[建立角色以委派許可給 AWS 服務](#)。

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

Note

如果您的 AWS 帳戶 已有 IAM AWSServiceRoleForEC2Spot 角色，則會收到下列錯誤。

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

AWS PCS 的最低許可

本節說明 IAM 身分（使用者、群組或角色）使用服務所需的最低 IAM 許可。

內容

- [使用 API 動作的最低許可](#)
- [使用標籤的最低許可](#)
- [支援日誌的最低許可](#)
- [使用容量區塊的最小許可](#)
- [服務管理員的最低許可](#)

使用 API 動作的最低許可

API 動作	最低許可	主控台的其他許可
CreateCluster	<pre>ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:GetSecurityGroupsForVpc, iam:CreateServiceLinkedRole, secretsmanager:CreateSecret, secretsmanager:TagResource, secretsmanager:RotateSecret,</pre>	

API 動作	最低許可	主控台的其他許可
	<code>pcs:CreateCluster</code>	
ListClusters	<code>pcs:ListClusters</code>	
GetCluster	<code>pcs:GetCluster</code>	<code>ec2:DescribeSubnets</code>
DeleteCluster	<code>pcs>DeleteCluster</code>	
CreateComputeNodeGroup	<code>ec2:DescribeVpcs,</code> <code>ec2:DescribeSubnets,</code> <code>ec2:DescribeSecurityGroups,</code> <code>ec2:DescribeLaunchTemplates,</code> <code>ec2:DescribeLaunchTemplateVersions,</code> <code>ec2:DescribeInstanceTypes,</code> <code>ec2:DescribeInstanceTypeOfferings,</code> <code>ec2:RunInstances,</code> <code>ec2:CreateFleet,</code> <code>ec2:CreateTags,</code> <code>iam:PassRole,</code> <code>iam:GetInstanceProfile,</code> <code>pcs:CreateComputeNodeGroup</code>	<code>iam:ListInstanceProfiles,</code> <code>ec2:DescribeImages,</code> <code>pcs:GetCluster</code>
ListComputerNodeGroups	<code>pcs:ListComputeNodeGroups</code>	<code>pcs:GetCluster</code>
GetComputeNodeGroup	<code>pcs:GetComputeNodeGroup</code>	<code>ec2:DescribeSubnets</code>

API 動作	最低許可	主控台的其他許可
UpdateComputeNodeGroup	<pre>ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:DescribeInstanceTypeOfferings, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags, iam:PassRole, iam:GetInstanceProfile, pcs:UpdateComputeNodeGroup</pre>	<pre>pcs:GetComputeNodeGroup, iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster</pre>
DeleteComputeNodeGroup	<pre>pcs>DeleteComputeNodeGroup</pre>	
CreateQueue	<pre>pcs>CreateQueue</pre>	<pre>pcs:ListComputeNodeGroups, pcs:GetCluster</pre>
ListQueues	<pre>pcs:ListQueues</pre>	<pre>pcs:GetCluster</pre>
GetQueue	<pre>pcs:GetQueue</pre>	
UpdateQueue	<pre>pcs:UpdateQueue</pre>	<pre>pcs:ListComputeNodeGroups, pcs:GetQueue</pre>

API 動作	最低許可	主控台的其他許可
DeleteQueue	<code>pcs:DeleteQueue</code>	

使用標籤的最低許可

在 AWS PCS 中將標籤與資源搭配使用需要下列許可。

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

支援日誌的最低許可

AWS PCS 會將日誌資料傳送至 Amazon CloudWatch Logs (CloudWatch Logs)。您必須確定您的身分具有使用 CloudWatch Logs 的最低許可。如需詳細資訊，請參閱《Amazon [CloudWatch Logs 使用者指南](#)》中的[管理 CloudWatch Logs 資源存取許可的概觀](#)。Amazon CloudWatch

如需有關服務將日誌傳送至 CloudWatch Logs 所需的許可資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的[啟用來自 AWS 服務的日誌記錄](#)。

使用容量區塊的最小許可

Amazon EC2 Capacity Blocks for ML 是一種 Amazon EC2 購買選項，可讓您預先付費，在特定日期和時間範圍內保留 GPU 加速運算執行個體，以支援短期工作負載。如需詳細資訊，請參閱[搭配 AWS PCS 使用適用於 ML 的 Amazon EC2 容量區塊](#)。

當您建立或更新運算節點群組時，您可以選擇使用容量區塊。您用來建立或更新運算節點群組的 IAM 身分必須具有下列許可：

```
ec2:DescribeCapacityReservations
```

服務管理員的最低許可

下列 IAM 政策指定 IAM 身分（使用者、群組或角色）設定和管理 AWS PCS 服務所需的最低許可。

Note

不設定和管理服務的使用者不需要這些許可。僅執行任務的使用者會使用安全殼層 (SSH) 連線到叢集。AWS Identity and Access Management (IAM) 不會處理 SSH 的身分驗證或授權。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PCSAccess",
      "Effect": "Allow",
      "Action": [
        "pcs:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2Access",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeImages",
        "ec2:GetSecurityGroupsForVpc",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:RunInstances",
        "ec2:CreateFleet",
        "ec2:CreateTags",
        "ec2:DescribeCapacityReservations"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamInstanceProfile",
      "Effect": "Allow",
      "Action": [
```

```

    "iam:GetInstanceProfile"
  ],
  "Resource": "*"
},
{
  "Sid": "IamPassRole",
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/*/AWSPCS*",
    "arn:aws:iam::*:role/AWSPCS*",
    "arn:aws:iam::*:role/aws-pcs/*",
    "arn:aws:iam::*:role/*/aws-pcs/*"
  ],
  "Condition": {
    "StringEquals": {
      "iam:PassedToService": [
        "ec2.amazonaws.com"
      ]
    }
  }
},
{
  "Sid": "SLRAccess",
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleFor*",
    "arn:aws:iam::*:role/aws-service-role/spot.amazonaws.com/AWSServiceRoleFor*"
  ],
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": [
        "pcs.amazonaws.com",
        "spot.amazonaws.com"
      ]
    }
  }
},
{

```

```

    "Sid": "AccessKMSKey",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
{
    "Sid": "SecretManagementAccess",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:CreateSecret",
        "secretsmanager:TagResource",
        "secretsmanager:UpdateSecret",
        "secretsmanager:RotateSecret"
    ],
    "Resource": "*"
},
{
    "Sid": "ServiceLogsDelivery",
    "Effect": "Allow",
    "Action": [
        "pcs:AllowVendedLogDeliveryForResource",
        "logs:PutDeliverySource",
        "logs:PutDeliveryDestination",
        "logs:CreateDelivery"
    ],
    "Resource": "*"
}
]
}

```

平行運算服務的 IAM AWS 執行個體描述檔

在 EC2 執行個體上執行的應用程式必須在其提出的任何 AWS API 請求中包含 AWS 登入資料。我們建議您使用 IAM 角色來管理 EC2 執行個體上的臨時登入資料。您可以定義執行個體描述檔來執行此操作，並將其連接到您的執行個體。如需詳細資訊，請參閱 [《Amazon Elastic Compute Cloud 使用者指南》](#) 中的 [Amazon EC2 的 IAM 角色](#)。

Note

當您使用 AWS 管理主控台 為 Amazon EC2 建立 IAM 角色時，主控台會自動建立執行個體描述檔，並為其提供與 IAM 角色相同的名稱。如果您使用 AWS CLI、AWS API 動作或 AWS SDK 來建立 IAM 角色，您可以將執行個體描述檔建立為個別動作。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的[執行個體描述檔](#)。

建立運算節點群組時，您必須指定執行個體描述檔的 Amazon Resource Name (ARN)。您可以為部分或全部運算節點群組選擇不同的執行個體描述檔。

要求

執行個體描述檔的 IAM 角色

與執行個體描述檔相關聯的 IAM 角色必須在其路徑/`aws-pcs/`中具有，或其名稱必須以開頭`AWSPCS`。

IAM 角色 ARNs 範例

- `arn:aws:iam::*:role/AWSPCS-example-role-1`
- `arn:aws:iam::*:role/aws-pcs/example-role-2`

許可

與 AWS PCS 執行個體描述檔相關聯的 IAM 角色必須包含下列政策。

JSON

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

```
]
}
```

其他政策

考慮將受管政策新增至執行個體描述檔。例如：

- [AmazonS3ReadOnlyAccess](#) 提供所有 S3 儲存貯體的唯讀存取權。
- [AmazonSSMManagedInstanceCore](#) 啟用 AWS Systems Manager 服務核心功能，例如直接從 Amazon 管理主控台進行遠端存取。
- [CloudWatchAgentServerPolicy](#) 包含在伺服器上使用 AmazonCloudWatchAgent 所需的許可。

您也可以包含自己的 IAM 政策，以支援您的特定使用案例。

建立 AWS PCS 的執行個體描述檔

AWS PCS console

當您建立運算節點群組，讓 AWS PCS 使用最低必要政策為您建立基本設定檔時，請選取建立設定檔。

Amazon EC2 console

您可以直接從 Amazon EC2 主控台建立執行個體描述檔。如需詳細資訊，請參閱AWS Identity and Access Management 《使用者指南》中的[使用執行個體描述檔](#)。

Important

請務必在 IAM 角色名稱AWSPCS中使用必要的字首。

AWS CLI

使用 AWS CLI 設定基本執行個體描述檔

Note

將下列範例中的 *example-role* 取代為您的 IAM 角色名稱。

1. 使用 `/aws-pcs/` 作為路徑屬性或開頭為 的名稱來建立 IAM 角色 `AWSPCS`。
 - a. 將下列內容複製並貼到名為 的新文字檔案 `trust_policy.json`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ec2.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}
```

- b. 使用下列 1 個命令來建立 IAM 角色。

```
aws iam create-role --path /aws-pcs/ --role-name example-role --assume-role-policy-document file://trust_policy.json
```

或

```
aws iam create-role --role-name AWSPCS-example-role --assume-role-policy-document file://trust_policy.json
```

2. 連接許可。
 - a. 將下列內容複製並貼到名為 的新文字檔案 `policy_document.json`。

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

```

        ],
        "Resource": "*",
        "Effect": "Allow"
    }
]
}

```

- b. 將政策文件連接至角色。此命令會將政策附加為內嵌政策。

```

aws iam put-role-policy \
  --role-name example-role \
  --policy-name pcsRegisterInstancePolicy \
  --policy-document file://policy_document.json

```

3. 建立執行個體描述檔。將 *example-profile* 取代為您的執行個體描述檔名稱。

```

aws iam create-instance-profile --instance-profile-name example-profile

```

4. 將 IAM 角色與執行個體描述檔建立關聯。

```

aws iam add-role-to-instance-profile \
  --instance-profile-name example-profile \
  --role-name example-role

```

尋找與 AWS PCS 搭配使用的執行個體描述檔

1. 如果您不知道 AWS PCS 的 IAM 角色的確切名稱，請使用下列 AWS CLI 命令列出符合 AWS PCS 名稱要求的 IAM 角色。

```

aws iam list-roles --query "Roles[?starts_with(RoleName, 'AWSPCS') ||
contains(Path, '/aws-pcs/)].[RoleName]" --output text

```

2. 使用以下 AWS CLI 命令列出與特定 IAM 角色相關聯的執行個體描述檔。將 *role-name* 取代為符合 AWS PCS 名稱需求的 IAM 角色名稱。

```

aws iam list-instance-profiles-for-role --role-name role-name

```

平行 AWS 運算服務身分和存取的故障診斷

使用以下資訊來協助您診斷和修正使用 AWS PCS 和 IAM 時可能遇到的常見問題。

主題

- [我無權在 AWS PCS 中執行動作](#)
- [我未獲得執行 iam:PassRole 的授權](#)
- [我想要允許以外的人員 AWS 帳戶 存取我的 AWS PCS 資源](#)

我無權在 AWS PCS 中執行動作

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在 mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 pcs:*GetWidget* 許可時發生。

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

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 pcs:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 iam:PassRole 動作，您的政策必須更新，以允許您將角色傳遞至 AWS PCS。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM marymajor 使用者嘗試使用主控台在 AWS PCS 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞給服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許以外的人員 AWS 帳戶 存取我的 AWS PCS 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 AWS PCS 是否支援這些功能，請參閱 [AWS Parallel Computing Service 如何與 IAM 搭配使用](#)。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《[IAM 使用者指南](#)》中的 [在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權](#)。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《[IAM 使用者指南](#)》中的 [將存取權提供給第三方 AWS 帳戶 擁有](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱《[IAM 使用者指南](#)》中的 [將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《[IAM 使用者指南](#)》中的 [IAM 中的跨帳戶資源存取](#)。

AWS 平行運算服務的合規驗證

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱[AWS 服務 合規計劃範圍內](#)然後選擇您感興趣的合規計劃。如需一般資訊，請參閱[AWS 合規計劃](#)。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載報告 in AWS Artifact](#)

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 [AWS 安全文件](#)。

AWS 平行運算服務的彈性

AWS 全球基礎設施是以 AWS 區域 和 可用區域為基礎建置。AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

AWS 平行運算服務中的基礎設施安全性

作為受管服務，AWS Parallel Computing Service 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱[AWS 雲端安全](#)。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的[基礎設施保護](#)。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 AWS PCS。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

AWS PCS 建立叢集時，服務會在服務擁有的帳戶中啟動 Slurm 控制器，與您帳戶中的運算節點分開。為了橋接控制器與運算節點之間的通訊，AWS PCS 會在您的 VPC 中建立跨帳戶彈性網路界面 (ENI)。Slurm 控制器使用 ENI 來管理和與不同的運算節點通訊 AWS 帳戶，維護資源的安全性和隔離，同時促進高效的 HPC 和 AI/ML 操作。

Parallel Computing Service AWS 中的漏洞分析和管理的

組態和 IT 控制是 AWS 與您之間共同責任。如需詳細資訊，請參閱 [AWS 共同責任模型](#)。AWS 會處理服務帳戶中基礎基礎設施的基本安全任務，例如在控制器執行個體上修補作業系統、防火牆組態和 AWS 基礎設施災難復原。這些程序已由適當的第三方進行檢閱並認證。如需詳細資訊，請參閱[安全性、身分和合規的最佳實務](#)。

Note

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

您要負責您 中基礎基礎設施的安全性 AWS 帳戶：

- 維護您的程式碼，包括更新和安全性修補程式。
- 修補和更新運算節點群組 Amazon Machine Image (AMI) 中的作業系統，並更新運算節點群組以使用更新的 AMI。

- 更新排程器，使其保持在支援的版本內。更新運算節點群組的 AMI，並更新運算節點群組以使用更新的 AMI。
- 驗證和加密使用者用戶端與其連線節點之間的通訊。

如需更新運算節點群組 AMI 的詳細資訊，請參閱 [AWS PCS 的 Amazon Machine Image AMIs](#)。

預防跨服務混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多權限的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況，AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

我們建議在資源政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容金鑰，以限制 AWS 平行運算服務 (AWS PCS) 為資源提供另一項服務的許可。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 `aws:SourceArn`。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

防範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容索引鍵，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 `aws:SourceArn` 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 `arn:aws:service:*:123456789012:*`。

如果 `aws:SourceArn` 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須使用這兩個全域條件內容索引鍵來限制許可。

的值 `aws:SourceArn` 必須是叢集 ARN。

下列範例示範如何在 AWS PCS 中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容金鑰，以防止混淆代理人問題。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "pcs.amazonaws.com"
    }
  }
}
```

```

    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": [
          "arn:aws:pcs:us-east-1:123456789012:cluster/*"
        ]
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
}
}

```

作為運算節點群組一部分佈建之 Amazon EC2 執行個體的 IAM 角色

AWS PCS 會自動協調叢集中每個已設定運算節點群組的 Amazon EC2 容量。建立運算節點群組時，使用者必須透過 `iamInstanceProfileArn` 欄位提供 IAM 執行個體描述檔。執行個體描述檔指定與佈建的 EC2 執行個體相關聯的許可。AWS PCS 接受具有 AWSPCS 做為角色名稱字首或 `/aws-pcs/` 做為角色路徑一部分的任何角色。建立或更新運算節點群組的 IAM 身分（使用者或角色）需要 `iam:PassRole` 許可。當使用者呼叫 `CreateComputeNodeGroup` 或 `UpdateComputeNodeGroup` API 動作時，AWS PCS 會檢查使用者是否可以執行 `iam:PassRole` 動作。

下列政策範例會授與僅傳遞名稱開頭為 AWSPCS 之 IAM 角色的許可。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/AWSPCS*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ec2.amazonaws.com"
          ]
        }
      }
    }
  ]
}

```

}

AWS Parallel Computing Service 的安全最佳實務

本節說明 AWS 平行運算服務 (AWS PCS) 特有的安全最佳實務。若要進一步了解 中的安全最佳實務 AWS，請參閱[安全性、身分和合規的最佳實務](#)。

AMI 相關安全性

- 請勿將 AWS PCS 範例 AMIs 用於生產工作負載。範例 AMIs 不受支援，且僅用於測試。
- 定期更新運算節點群組 AMI 中的作業系統和軟體，以緩解漏洞。
- 僅使用從官方 AWS 來源下載的已驗證官方 AWS PCS 套件。
- 定期更新運算節點群組 AMI 中的 AWS PCS 套件，並更新運算節點以使用更新的 AMI。請考慮自動化此程序，以將漏洞降至最低。

如需詳細資訊，請參閱[AWS PCS 的自訂 Amazon Machine Image AMIs](#)。

Slurm Workload Manager 安全性

- 實作存取控制和網路限制，以保護 Slurm 控制和運算節點。僅允許信任的使用者和系統提交任務和存取 Slurm 管理命令。
- 使用 Slurm 的內建安全功能，例如 Slurm 身分驗證，以確保任務提交和通訊已進行身分驗證。
- 更新 Slurm 版本以維持順暢的操作和叢集支援。

Important

任何使用已達到支援生命週期 (EOSL) 結束之 Slurm 版本的叢集都會立即停止。使用使用者指南頁面頂端的連結來訂閱 AWS PCS 文件 RSS 摘要，以在 Slurm 版本接近 EOSL 時收到通知。

如需詳細資訊，請參閱[AWS PCS 中的 Slurm 版本](#)。

- 定期輪換叢集秘密，以維持安全合規並修復潛在的安全漏洞。這是 HIPAA 和 FedRAMP 合規的必要項目。

如需詳細資訊，請參閱[在 AWS PCS 中輪換叢集秘密](#)。

監控和記錄

- 使用 Amazon CloudWatch Logs 和 AWS CloudTrail 來監控和記錄叢集和 中的動作 AWS 帳戶。使用資料進行疑難排解和稽核。

網路安全

- 在不同的 VPC 中部署 AWS PCS 叢集，以隔離您的 HPC 環境與其他網路流量。
- 使用安全群組和網路存取控制清單 (ACLs) 來控制 AWS PCS 執行個體和子網路的傳入和傳出流量。
- 使用 AWS PrivateLink 或 VPC 端點來保持叢集與網路 AWS 內其他服務之間的 AWS 網路流量。如需詳細資訊，請參閱[AWS Parallel Computing 服務 使用界面端點存取 \(AWS PrivateLink\)](#)。

AWS PCS 的記錄和監控

監控是維護 AWS PCS 和其他 AWS 資源可靠性、可用性和效能的重要部分。AWS 提供下列監控工具來監看 AWS PCS、在發生錯誤時回報，以及適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。
- Amazon CloudWatch Logs 可讓您監控、存放和存取來自 Amazon EC2 執行個體、CloudTrail 及其他來源的日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon CloudWatch Logs 使用者指南](#)。
- AWS CloudTrail 會擷取您 AWS 帳戶或代表您的帳戶發出的 API 呼叫和相關事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱 [「AWS CloudTrail 使用者指南」](#)。

AWS PCS 中的任務完成日誌

任務完成日誌提供您平行 AWS 運算服務 (AWS PCS) 任務完成時的金鑰詳細資訊，無需額外費用。您可以使用其他 AWS 服務來存取和處理您的日誌資料，例如 Amazon CloudWatch Logs、Amazon Simple Storage Service (Amazon S3) 和 Amazon Data Firehose；AWS PCS 會記錄任務的中繼資料，例如下列項目。

- 任務 ID 和名稱
- 使用者和群組資訊
- 任務狀態 (例如 COMPLETED、FAILED、CANCELLED)
- 使用的分割區
- 時間限制
- 開始、結束、提交和合格時間
- 節點清單和計數
- 處理器計數
- 工作目錄
- 資源用量 (CPU、記憶體)

- 結束代碼
- 節點詳細資訊 (名稱、執行個體 IDs、執行個體類型)

內容

- [先決條件](#)
- [設定任務完成日誌](#)
- [如何尋找任務完成日誌](#)
 - [CloudWatch Logs](#)
 - [Amazon S3](#)
- [任務完成日誌欄位](#)
- [任務完成日誌範例](#)

先決條件

管理 AWS PCS 叢集的 IAM 主體必須允許 `pcs:AllowVendedLogDeliveryForResource` 動作。

下列範例 IAM 政策會授予必要的許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PcsAllowVendedLogsDelivery",
      "Effect": "Allow",
      "Action": ["pcs:AllowVendedLogDeliveryForResource"],
      "Resource": [
        "arn:aws:pcs:*::cluster/*"
      ]
    }
  ]
}
```

設定任務完成日誌

您可以使用 AWS 管理主控台 或 為您的 AWS PCS 叢集設定任務完成日誌 AWS CLI。

AWS 管理主控台

使用主控台設定任務完成日誌

1. 開啟 [AWS PCS 主控台](#)。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要新增任務完成日誌的叢集。
4. 在叢集詳細資訊頁面上，選擇日誌索引標籤。
5. 在任務完成日誌下，選擇新增以在 CloudWatch Logs、Amazon S3 和 Firehose 之間新增最多 3 個日誌交付目的地。
6. 選擇更新日誌交付。

AWS CLI

使用 設定任務完成日誌 AWS CLI

1. 建立日誌交付目的地：

```
aws logs put-delivery-destination --region region \  
  --name pcs-logs-destination \  
  --delivery-destination-configuration \  
  destinationResourceArn=resource-arn
```

取代：

- *region* — 您要建立目的地 AWS 區域的，例如 us-east-1
- *pcs-logs-destination* — 目的地的名稱
- *resource-arn* — CloudWatch Logs 日誌群組、S3 儲存貯體或 Firehose 交付串流的 Amazon Resource Name (ARN)。

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考指南》中的 [PutDeliveryDestination](#)。

2. 將 PCS 叢集設定為日誌交付來源：

```
aws logs put-delivery-source --region region \  
  --name cluster-logs-source-name \  
  --resource-arn cluster-arn \  
  --log-type PCS_JOBCOMP_LOGS
```

取代：

- *region* — 叢集 AWS 區域的，例如 us-east-1
- *cluster-logs-source-name* — 來源的名稱
- *cluster-arn* — AWS PCS 叢集的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 [PutDeliverySource](#)。

3. 將交付來源連接至交付目的地：

```
aws logs create-delivery --region region \  
  --delivery-source-name cluster-logs-source \  
  --delivery-destination-arn destination-arn
```

取代：

- *region* — AWS 區域，例如 us-east-1
- *cluster-logs-source* — 交付來源的名稱
- *destination-arn* — 交付目的地的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 [CreateDelivery](#)。

如何尋找任務完成日誌

您可以在 CloudWatch Logs 和 Amazon S3. AWS PCS 中設定日誌目的地，並使用下列結構化路徑名稱和檔案名稱。

CloudWatch Logs

AWS PCS 使用 CloudWatch Logs 串流的下列名稱格式：

```
AWSLogs/PCS/cluster-id/jobcomp.log
```

例如：AWSLogs/PCS/pcs_abc123de45/jobcomp.log

Amazon S3

AWS PCS 使用 S3 路徑的下列名稱格式：

```
AWSLogs/account-id/PCS/region/cluster-id/jobcomp/year/month/day/hour/
```

例如：AWSLogs/111122223333/PCS/us-east-1/pcs_abc123de45/
jobcomp/2025/06/19/11/

AWS PCS 對日誌檔案使用以下名稱格式：

```
PCS_jobcomp_year-month-day-hour_cluster-id_random-id.log.gz
```

例如：PCS_jobcomp_2025-06-19-11_pcs_abc123de45_04be080b.log.gz

任務完成日誌欄位

AWS PCS 會將任務完成日誌資料寫入為 JSON 物件。JSON 容器 jobcomp 會保留任務詳細資訊。下表說明 jobcomp 容器內的欄位。某些欄位只會在特定情況下存在，例如陣列任務或異質任務。

任務完成日誌欄位

名稱	範例值	必要	備註
job_id	11	是	永遠具有值
user	"root"	是	永遠具有值
user_id	0	是	永遠具有值
group	"root"	是	永遠具有值
group_id	0	是	永遠具有值
name	"wrap"	是	永遠具有值
job_state	"COMPLETED"	是	永遠具有值

名稱	範例值	必要	備註
partition	"Hydra-Mp iQueue-ab cdef01-7"	是	永遠具有 值
time_limit	"UNLIMITED"	是	永遠存在，但可能是 "UNLIMITED"
start_time	"2025-06- 19T10:58: 57"	是	永遠存在，但可能是 "Unknown"
end_time	"2025-06- 19T10:58: 57"	是	永遠存在，但可能是 "Unknown"
node_list	"Hydra-Mp iNG-abcde f01-2345- 1"	是	永遠具有 值
node_cnt	1	是	永遠具有 值
proc_cnt	1	是	永遠具有 值
work_dir	"/root"	是	永遠存在，但可能是 "Unknown"
reservati on_name	"weekly_m aintenanc e"	是	永遠存在，但可能是空字串 ""
tres.cpu	1	是	永遠具有 值
tres.mem. val	600	是	永遠具有 值
tres.mem. unit	"M"	是	可以是 "M"或 "bb"
tres.node	1	是	永遠具有 值

名稱	範例值	必要	備註
tres.billing	1	是	永遠具有值
account	"finance"	是	永遠存在，但可能是空字串 ""
qos	"normal"	是	永遠存在，但可能是空字串 ""
wc_key	"project_1"	是	永遠存在，但可能是空字串 ""
cluster	"unknown"	是	永遠存在，但可能是 "unknown"
submit_time	"2025-06-19T10:55:46"	是	永遠存在，但可能是 "Unknown"
eligible_time	"2025-06-19T10:55:46"	是	永遠存在，但可能是 "Unknown"
array_job_id	12	編號	只有在任務是陣列任務時才會出現
array_task_id	1	編號	只有在任務是陣列任務時才會出現
het_job_id	10	編號	只有在任務是異質任務時才會出現
het_job_offset	0	編號	只有在任務是異質任務時才會出現
derived_exit_code_status	0	是	永遠具有值
derived_exit_code_signal	0	是	永遠具有值

名稱	範例值	必要	備註
exit_code_status	0	是	永遠具有值
exit_code_signal	0	是	永遠具有值
node_details[0].name	"Hydra-MpiNG-abcdef01-2345-1"	編號	永遠存在，但node_details 可能是 "[]"
node_details[0].instance_id	"i-0abcdef01234567a"	編號	永遠存在，但node_details 可能是 "[]"
node_details[0].instance_type	"t4g.micro"	編號	永遠存在，但node_details 可能是 "[]"

任務完成日誌範例

下列範例顯示各種任務類型和狀態的任務完成日誌：

```
{ "jobcomp": { "job_id": 1, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:32:57", "end_time": "2025-06-19T16:33:03", "node_list": "Hydra-MpiNG-abcdef01-2345-[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name": "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2, "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T16:29:40", "eligible_time": "2025-06-19T16:29:41", "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name": "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type": "t4g.micro" } ] } }
```

```

{ "jobcomp": { "job_id": 2, "user": "root", "user_id": 0, "group": "root", "group_id":
  0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-
  abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:33:13",
  "end_time": "2025-06-19T16:33:14", "node_list": "Hydra-MpiNG-abcdef01-2345-
  [1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name":
  "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
  "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
  "submit_time": "2025-06-19T16:33:13", "eligible_time": "2025-06-19T16:33:13",
  "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
  0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
  "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name":
  "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type":
  "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 3, "user": "root", "user_id": 0, "group": "root", "group_id":
  0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
  "time_limit": "UNLIMITED", "start_time": "2025-06-19T22:58:57", "end_time":
  "2025-06-19T22:58:57", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
  1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
  1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
  "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T22:55:46",
  "eligible_time": "2025-06-19T22:55:46", "derived_exit_code_status": 0,
  "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal":
  0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id":
  "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 4, "user": "root", "user_id": 0, "group": "root",
  "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-
  MpiQueue-abcdef01-7", "time_limit": "525600", "start_time": "2025-06-19T23:04:27",
  "end_time": "2025-06-19T23:04:27", "node_list": "Hydra-MpiNG-abcdef01-2345-
  [1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/root", "reservation_name":
  "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
  "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
  "submit_time": "2025-06-19T23:01:38", "eligible_time": "2025-06-19T23:01:38",
  "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
  0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
  "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" }, { "name":
  "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def345abc67890", "instance_type":
  "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 5, "user": "root", "user_id": 0, "group": "root", "group_id":
  0, "name": "wrap", "job_state": "FAILED", "partition": "Hydra-MpiQueue-abcdef01-7",
  "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:00", "end_time":
  "2025-06-19T23:09:00", "node_list": "(null)", "node_cnt": 0, "proc_cnt": 0,
  "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val":
  1, "unit": "G" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key":
  "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:00", "eligible_time":

```

```

"2025-06-19T23:09:00", "derived_exit_code_status": 0, "derived_exit_code_signal": 0,
"exit_code_status": 0, "exit_code_signal": 1, "node_details": [ ] } }
{ "jobcomp": { "job_id": 6, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:36",
"end_time": "2025-06-19T23:09:36", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
{ "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
"", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:35",
"eligible_time": "2025-06-19T23:09:36", "het_job_id": 6, "het_job_offset": 0,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
"exit_code_signal": 1, "node_details": [ ] } }
{ "jobcomp": { "job_id": 7, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:10:03",
"end_time": "2025-06-19T23:10:03", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
{ "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
"", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:10:03",
"eligible_time": "2025-06-19T23:10:03", "het_job_id": 7, "het_job_offset": 0,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
"exit_code_signal": 1, "node_details": [ ] } }
{ "jobcomp": { "job_id": 8, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
"2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
"eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 0,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
"instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 9, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
"2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
"eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 1,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
"instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }

```

```

{ "jobcomp": { "job_id": 10, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
"2025-06-19T23:12:24", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
"eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 0,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
"instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 11, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
"2025-06-19T23:12:24", "node_list": "Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 600, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
"eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 1,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
"instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 13, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:57", "end_time":
"2025-06-19T23:47:58", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
"eligible_time": "2025-06-19T23:43:56", "array_job_id": 12, "array_task_id": 1,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
"instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 12, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:58", "end_time":
"2025-06-19T23:47:58", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
"eligible_time": "2025-06-19T23:43:56", "array_job_id": 12, "array_task_id": 2,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":

```

```
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",  
"instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
```

AWS PCS 中的排程器日誌

您可以設定 AWS PCS 將詳細的記錄資料從叢集排程器傳送至 Amazon CloudWatch Logs、Amazon Simple Storage Service (Amazon S3) 和 Amazon Data Firehose。這可協助監控和故障診斷。

內容

- [先決條件](#)
- [設定排程器日誌](#)
- [排程器日誌串流路徑和名稱](#)
- [排程器日誌記錄範例](#)

先決條件

管理 AWS PCS 叢集的 IAM 主體必須允許 `pcs:AllowVendedLogDeliveryForResource` 動作。

下列範例 IAM 政策會授予必要的許可。

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PcsAllowVendedLogsDelivery",  
      "Effect": "Allow",  
      "Action": ["pcs:AllowVendedLogDeliveryForResource"],  
      "Resource": [  
        "arn:aws:pcs:*:*:cluster/*"  
      ]  
    }  
  ]  
}
```

設定排程器日誌

您可以使用 AWS 管理主控台 或 為您的 AWS PCS 叢集設定排程器日誌 AWS CLI。

AWS 管理主控台

使用主控台設定排程器日誌

1. 開啟 [AWS PCS 主控台](#)。
2. 在導覽窗格中，選擇叢集。
3. 選擇您要新增排程器日誌的叢集。
4. 在叢集詳細資訊頁面上，選擇日誌索引標籤。
5. 在排程器日誌下，選擇新增以在 CloudWatch Logs、Amazon S3 和 Firehose 之間新增最多 3 個日誌交付目的地。
6. 選擇更新日誌交付。

AWS CLI

使用 設定排程器日誌 AWS CLI

1. 建立日誌交付目的地：

```
aws logs put-delivery-destination --region region \  
  --name pcs-logs-destination \  
  --delivery-destination-configuration \  
  destinationResourceArn=resource-arn
```

取代：

- *region* — 您要建立目的地 AWS 區域的，例如 us-east-1
- *pcs-logs-destination* — 目的地的名稱
- *resource-arn* — CloudWatch Logs 日誌群組、S3 儲存貯體或 Firehose 交付串流的 Amazon Resource Name (ARN)。

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考指南》中的 [PutDeliveryDestination](#)。

2. 將 PCS 叢集設定為日誌交付來源：

```
aws logs put-delivery-source --region region \  
  --name cluster-logs-source-name \  
  --resource-arn cluster-arn \  
  --log-type PCS_SCHEDULER_LOGS
```

取代：

- *region* — 叢集 AWS 區域的，例如 us-east-1
- *cluster-logs-source-name* — 來源的名稱
- *cluster-arn* — AWS PCS 叢集的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 [PutDeliverySource](#)。

3. 將交付來源連接至交付目的地：

```
aws logs create-delivery --region region \  
  --delivery-source-name cluster-logs-source \  
  --delivery-destination-arn destination-arn
```

取代：

- *region* — AWS 區域，例如 us-east-1
- *cluster-logs-source* — 交付來源的名稱
- *destination-arn* — 交付目的地的 ARN

如需詳細資訊，請參閱《Amazon CloudWatch Logs API 參考》中的 [CreateDelivery](#)。

排程器日誌串流路徑和名稱

AWS PCS 排程器日誌的路徑和名稱取決於目的地類型。

- CloudWatch Logs
 - CloudWatch Logs 串流遵循此命名慣例。

```
AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
```

Example

```
AWSLogs/PCS/abcdef0123/slurmctld_24.05.log
```

- S3 bucket (S3 儲存貯體)
- S3 儲存貯體輸出路徑遵循此命名慣例：

```
AWSLogs/${account-id}/PCS/${region}/${cluster_id}/${log_name}/
${scheduler_major_version}/yyyy/MM/dd/HH/
```

Example

```
AWSLogs/111111111111/PCS/us-east-2/abcdef0123/slurmctld/24.05/2024/09/01/00.
```

- S3 物件名稱遵循此慣例：

```
PCS_${log_name}_${scheduler_major_version}_#{expr date 'event_timestamp', format:
"yyyy-MM-dd-HH"}_${cluster_id}_${hash}.log
```

Example

```
PCS_slurmctld_24.05_2024-09-01-00_abcdef0123_0123abcdef.log
```

排程器日誌記錄範例

AWS PCS 排程器日誌是結構化的。除了從 Slurm 控制器程序發出的日誌訊息之外，還包含叢集識別符、排程器類型、主要和修補程式版本等欄位。請見此處範例。

```
{
  "resource_id": "s3431v9rx2",
  "resource_type": "PCS_CLUSTER",
  "event_timestamp": 1721230979,
  "log_level": "info",
  "log_name": "slurmctld",
  "scheduler_type": "slurm",
  "scheduler_major_version": "25.05",
  "scheduler_patch_version": "3",
  "node_type": "controller_primary",
```

```
"message": "[2024-07-17T15:42:58.614+00:00] Running as primary controller\n"
}
```

使用 Amazon CloudWatch 監控 AWS 平行運算服務

Amazon CloudWatch 會定期從叢集收集指標，藉此監控 AWS 您的平行運算服務 (AWS PCS) 叢集運作狀態和效能。這些指標會保留，可讓您存取歷史資料，並隨著時間深入了解叢集的效能。

CloudWatch 也可讓您監控 AWS PCS 啟動的 EC2 執行個體，以符合您的擴展需求。雖然您可以檢查執行中執行個體的日誌，但 CloudWatch 指標和日誌記錄資料通常會在執行個體終止後刪除。不過，您可以使用 EC2 啟動範本在執行個體上設定 CloudWatch 代理程式，以便在執行個體終止後保留指標和日誌，進而進行長期監控和分析。

探索本節中的主題，進一步了解如何使用 CloudWatch 監控 AWS PCS。

主題

- [使用 CloudWatch 監控 AWS PCS 指標](#)
- [使用 Amazon CloudWatch 監控 AWS PCS 執行個體](#)

使用 CloudWatch 監控 AWS PCS 指標

您可以使用 Amazon CloudWatch 監控 AWS PCS 叢集運作狀態，這會從您的叢集收集資料，並將其轉換為近乎即時的指標。這些統計資料會保留 15 個月，讓您可以存取歷史資訊，並更清楚叢集的執行方式。叢集指標會以 1 分鐘的期間傳送至 CloudWatch。如需 CloudWatch 的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 [什麼是 Amazon CloudWatch ?](#)。

AWS PCS 會將下列指標發佈至 CloudWatch 中的 AWS/PCS 命名空間。它們具有單一維度 ClusterId。

名稱	描述	單位
ActualCapacity	IdleCapacity + UtilizedCapacity	計數
CapacityUtilization	UtilizedCapacity/ActualCapacity	計數

名稱	描述	單位
DesiredCapacity	ActualCapacity + PendingCapacity	計數
IdleCapacity	正在執行但未配置給任務的執行個體計數	計數
UtilizedCapacity	正在執行並配置給任務的執行個體計數	計數

使用 Amazon CloudWatch 監控 AWS PCS 執行個體

AWS PCS 會視需要啟動 Amazon EC2 執行個體，以符合 PCS 運算節點群組中定義的擴展需求。您可以在使用 Amazon CloudWatch 執行時監控這些執行個體。您可以透過登入並使用互動式命令列工具來檢查執行中執行個體的日誌。不過，根據預設，CloudWatch 指標資料只會在執行個體終止後保留一段有限期間，且執行個體日誌通常會與傳回執行個體的 EBS 磁碟區一起刪除。若要保留 PCS 在終止執行個體之後啟動的指標或記錄資料，您可以使用 EC2 啟動範本在執行個體上設定 CloudWatch 代理程式。本主題提供監控執行中執行個體的概觀，並提供如何設定持久性執行個體指標和日誌的範例。

監控執行中的執行個體

尋找 AWS PCS 執行個體

若要監控 PCS 啟動的執行個體，請尋找與叢集或運算節點群組相關聯的執行中執行個體。然後，在指定執行個體的 EC2 主控台中，檢查狀態和警示以及監控區段。如果已針對這些執行個體設定登入存取，您可以連線到這些執行個體，並檢查執行個體上的各種日誌檔案。如需識別哪些執行個體由 PCS 管理的詳細資訊，請參閱 [尋找 AWS PCS 中的運算節點群組執行個體](#)。

啟用詳細指標

根據預設，執行個體指標會以 5 分鐘的間隔收集。若要每隔一分鐘收集指標，請在運算節點群組啟動範本中啟用詳細的 CloudWatch 監控。如需詳細資訊，請參閱 [開啟詳細的 CloudWatch 監控](#)。

設定持久性執行個體指標和日誌

您可以在執行個體上安裝和設定 Amazon CloudWatch 代理程式，以保留其指標和日誌。這包含三個主要步驟：

1. 建立 CloudWatch 代理程式組態。

2. 將組態存放在 PCS 執行個體可以擷取的位置。
3. 撰寫 EC2 啟動範本，安裝 CloudWatch 代理程式軟體、擷取您的組態，並使用組態啟動 CloudWatch 代理程式。

如需詳細資訊，請參閱《Amazon [CloudWatch 使用者指南](#)》中的使用 [CloudWatch 代理程式收集指標、日誌和追蹤](#)，以及 [搭配 AWS PCS 使用 Amazon EC2 啟動範本](#)。Amazon CloudWatch

建立 CloudWatch Agent 組態

在執行個體上部署 CloudWatch 代理程式之前，您必須產生 JSON 組態檔案，指定要收集的指標、日誌和追蹤。您可以使用精靈或使用文字編輯器手動建立組態檔案。系統會為此示範手動建立組態檔案。

在已安裝 AWS CLI 的電腦上，使用下列內容建立名為 config.json 的 CloudWatch 組態檔案。您也可以使用下列 URL 下載檔案的副本。

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

備註

- 範例檔案中的日誌路徑適用於 Amazon Linux 2。如果您的執行個體將使用不同的基礎作業系統，請視需要變更路徑。
- 若要擷取其他日誌，請在下新增其他項目 collect_list。
- 中的值 {brackets} 是範本變數。如需支援變數的完整清單，請參閱《Amazon [CloudWatch 使用者指南](#)》中的 [手動建立或編輯 CloudWatch 代理程式組態檔案](#)。Amazon CloudWatch
- 您可以選擇省略 logs 或 metrics 如果您不想收集這些資訊類型。

```
{
  "agent": {
    "metrics_collection_interval": 60
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/var/log/cloud-init.log",
            "log_group_class": "STANDARD",
            "log_group_name": "/PCSLogs/instances",
            "log_stream_name": "{instance_id}.cloud-init.log",
```

```

        "retention_in_days": 30
    },
    {
        "file_path": "/var/log/cloud-init-output.log",
        "log_group_class": "STANDARD",
        "log_stream_name": "{instance_id}.cloud-init-output.log",
        "log_group_name": "/PCSLogs/instances",
        "retention_in_days": 30
    },
    {
        "file_path": "/var/log/amazon/pcs/bootstrap.log",
        "log_group_class": "STANDARD",
        "log_stream_name": "{instance_id}.bootstrap.log",
        "log_group_name": "/PCSLogs/instances",
        "retention_in_days": 30
    },
    {
        "file_path": "/var/log/slurmd.log",
        "log_group_class": "STANDARD",
        "log_stream_name": "{instance_id}.slurmd.log",
        "log_group_name": "/PCSLogs/instances",
        "retention_in_days": 30
    },
    {
        "file_path": "/var/log/messages",
        "log_group_class": "STANDARD",
        "log_stream_name": "{instance_id}.messages",
        "log_group_name": "/PCSLogs/instances",
        "retention_in_days": 30
    },
    {
        "file_path": "/var/log/secure",
        "log_group_class": "STANDARD",
        "log_stream_name": "{instance_id}.secure",
        "log_group_name": "/PCSLogs/instances",
        "retention_in_days": 30
    }
]
}
},
"metrics": {
    "aggregation_dimensions": [
        [

```

```
        "InstanceId"
      ]
    ],
    "append_dimensions": {
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "cpu": {
        "measurement": [
          "cpu_usage_idle",
          "cpu_usage_iowait",
          "cpu_usage_user",
          "cpu_usage_system"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ],
        "totalcpu": false
      },
      "disk": {
        "measurement": [
          "used_percent",
          "inodes_free"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      },
      "diskio": {
        "measurement": [
          "io_time"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      },
      "mem": {
        "measurement": [
```


首先，(如果您有現有的儲存貯體，這是選用的)，請建立儲存貯體以保留您的組態檔案(s)。

```
aws s3 mb s3://amzn-s3-demo-bucket
```

接著，將檔案上傳至儲存貯體。

```
aws s3 cp ./config.json s3://amzn-s3-demo-bucket/
```

儲存為 SSM 參數

若要將檔案儲存為 SSM 參數，請使用以下命令。在執行命令之前，請進行下列取代：

- 將 *region-code* 取代為您使用 AWS PCS 的 AWS 區域。
- (選用) 將 *AmazonCloudWatch-PCS* 取代為參數的自有名稱。請注意，如果您從變更名稱的字首 AmazonCloudWatch-，您將需要在節點群組執行個體描述檔中特別新增對 SSM 參數的讀取存取權。

```
aws ssm put-parameter \  
  --region region-code \  
  --name "AmazonCloudWatch-PCS" \  
  --type String \  
  --value file://config.json
```

撰寫 EC2 啟動範本

啟動範本的特定詳細資訊取決於您的組態檔案是存放在 S3 還是 SSM 中。

使用存放在 S3 中的組態

此指令碼會安裝 CloudWatch 代理程式、從 S3 儲存貯體匯入組態檔案，以及啟動 CloudWatch 代理程式。使用您自己的詳細資訊取代此指令碼中的下列值：

- *amzn-s3-demo-bucket* – 您的帳戶可從中讀取的 S3 儲存貯體名稱
- */config.json* – 相對於儲存組態之 S3 儲存貯體根目錄的路徑

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

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

packages:
- amazon-cloudwatch-agent

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/config.json /etc/s3-cw-config.json
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
  ec2 -s -c file://etc/s3-cw-config.json

---MYBOUNDARY---
```

節點群組的 IAM 執行個體描述檔必須能夠存取 儲存貯體。以下是上述使用者資料指令碼中儲存貯體的範例 IAM 政策。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

另請注意，執行個體必須允許 S3 和 CloudWatch 端點的傳出流量。這可以使用安全群組或 VPC 端點來完成，具體取決於您的叢集架構。

使用存放在 SSM 中的組態

此指令碼會安裝 CloudWatch 代理程式、從 SSM 參數匯入組態檔案，以及啟動 CloudWatch 代理程式。使用您自己的詳細資訊取代此指令碼中的下列值：

- (選用) 將 *AmazonCloudWatch-PCS* 取代為 參數的自有名稱。

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

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

packages:
- amazon-cloudwatch-agent

runcmd:
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
  ec2 -s -c ssm:AmazonCloudWatch-PCS

--==MYBOUNDARY==--
```

節點群組的 IAM 執行個體政策必須連接 CloudWatchAgentServerPolicy。

如果您的參數名稱開頭不是 AmazonCloudWatch-，您將需要在節點群組執行個體描述檔中特別新增對 SSM 參數的讀取存取權。以下是說明 *DOC-EXAMPLE-PREFIX* 字首的範例 IAM 政策。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CustomCwSsmMParamReadOnly",
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameter"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

另請注意，執行個體必須允許 SSM 和 CloudWatch 端點的傳出流量。這可以使用安全群組或 VPC 端點來完成，具體取決於您的叢集架構。

使用記錄 AWS 平行運算服務 API 呼叫 AWS CloudTrail

AWS PCS 已與整合 AWS CloudTrail，此服務提供使用者、角色或 AWS PCS AWS 服務所採取動作的記錄。CloudTrail 會將 AWS PCS 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 AWS PCS 主控台的呼叫，以及對 AWS PCS API 操作的程式碼呼叫。如果您建立線索，您可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 AWS PCS 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台的事件歷史記錄檢視最新事件。使用 CloudTrail 收集的資訊，您可以判斷對 AWS PCS 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [「AWS CloudTrail 使用者指南」](#)。

AWS CloudTrail 中的 PCS 資訊

建立帳戶 AWS 帳戶時，您的上會啟用 CloudTrail。當活動在 AWS PCS 中發生時，該活動會記錄於 CloudTrail 事件，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 [中](#)檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱 [「使用 CloudTrail 事件歷史記錄檢視事件」](#)。

若要持續記錄中的事件 AWS 帳戶，包括 AWS PCS 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。追蹤會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並對其採取行動。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案](#)

CloudTrail 會記錄所有 AWS PCS 動作，並記錄在[AWS 平行運算服務 API 參考](#)中。例如，對 CreateComputeNodeGroup、UpdateQueue 以及 DeleteCluster 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 請求是否使用根還是 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

從 AWS PCS 了解 CloudTrail 日誌檔案項目

權杖是一種組態，能讓事件以日誌檔案的形式交付至您指定的 S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

下列範例顯示 CreateQueue 動作的 CloudTrail 日誌項目。

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "ASIAY36PTPIEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAY36PTPIEEXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2024-07-16T17:05:51Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2024-07-16T17:13:09Z",
  "eventSource": "pcs.amazonaws.com",
  "eventName": "CreateQueue",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36",
  "requestParameters": {
    "clientToken": "c13b7baf-2894-42e8-acec-example",
    "clusterIdentifier": "abcdef0123",
```

```
    "computeNodeGroupConfigurations": [
      {
        "computeNodeGroupId": "abcdef0123"
      }
    ],
    "queueName": "all"
  },
  "responseElements": {
    "queue": {
      "arn": "arn:aws:pcs:us-east-1:609783872011:cluster/abcdef0123/queue/
abcdef0123",
      "clusterId": "abcdef0123",
      "computeNodeGroupConfigurations": [
        {
          "computeNodeGroupId": "abcdef0123"
        }
      ],
      "createdAt": "2024-07-16T17:13:09.276069393Z",
      "id": "abcdef0123",
      "modifiedAt": "2024-07-16T17:13:09.276069393Z",
      "name": "all",
      "status": "CREATING"
    }
  },
  "requestID": "a9df46d7-3f6d-43a0-9e3f-example",
  "eventID": "7ab18f88-0040-47f5-8388-example",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "012345678910",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "pcs.us-east-1.amazonaws.com"
  },
  "sessionCredentialFromConsole": "true"
}
```

AWS PCS 的端點和服務配額

下列各節說明 AWS 平行運算服務 (AWS PCS) 的端點和服務配額。服務配額，先前稱為限制，是的服務資源或操作數目上限 AWS 帳戶。

您的 AWS 帳戶 具有每個 AWS 服務的預設配額。除非另有說明，否則每個配額都是區域特定的。您可以請求提高某些配額，而其他配額無法提高。

如需詳細資訊，請參閱 AWS 一般參考中的 [AWS 服務配額](#)。

內容

- [服務端點](#)
- [Service Quotas](#)
 - [內部配額](#)
 - [其他服務的相關配額 AWS](#)

服務端點

區域名稱	區域	端點	通訊協定
美國東部 (俄亥俄)	us-east-2	pcs.us-east-2.amazonaws.com	HTTPS
		pcs-fips.us-east-2.amazonaws.com	
		pcs-fips.us-east-2.api.aws	
		pcs.us-east-2.api.aws	
美國東部 (維吉尼亞北部)	us-east-1	pcs.us-east-1.amazonaws.com	HTTPS
		pcs-fips.us-east-1.amazonaws.com	

區域名稱	區域	端點	通訊協定
		pcs-fips.us-east-1 .api.aws pcs.us-east-1.api.aws	
美國西部 (奧勒岡)	us-west-2	pcs.us-west-2.amaz onaws.com pcs-fips.us-west-2 .amazonaws.com pcs-fips.us-west-2 .api.aws pcs.us-west-2.api.aws	HTTPS
亞太區域 (孟買)	ap-south-1	pcs.ap-south-1.ama zonaws.com pcs.ap-south-1.api .aws	HTTPS
亞太區域 (新加坡)	ap-southeast-1	pcs.ap-southeast-1 .amazonaws.com pcs.ap-southeast-1 .api.aws	HTTPS
亞太地區 (悉尼)	ap-southeast-2	pcs.ap-southeast-2 .amazonaws.com pcs.ap-southeast-2 .api.aws	HTTPS
亞太地區 (東京)	ap-northeast-1	pcs.ap-northeast-1 .amazonaws.com pcs.ap-northeast-1 .api.aws	HTTPS

區域名稱	區域	端點	通訊協定
歐洲 (法蘭克福)	eu-central-1	pcs.eu-central-1.amazonaws.com pcs.eu-central-1.amazonaws.com	HTTPS
歐洲 (愛爾蘭)	eu-west-1	pcs.eu-west-1.amazonaws.com pcs.eu-west-1.amazonaws.com	HTTPS
歐洲 (倫敦)	eu-west-2	pcs.eu-west-2.amazonaws.com pcs.eu-west-2.amazonaws.com	HTTPS
Europe (Paris)	eu-west-3	pcs.eu-west-3.amazonaws.com pcs.eu-west-3.amazonaws.com	HTTPS
歐洲 (米蘭)	eu-south-1	pcs.eu-south-1.amazonaws.com pcs.eu-south-1.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	pcs.eu-north-1.amazonaws.com pcs.eu-north-1.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
AWS GovCloud (美國東部)	us-gov-east-1	pcs.us-gov-east-1. amazonaws.com	HTTPS
		pcs-fips.us-gov-east-1. amazonaws.com	
		pcs-fips.us-gov-east-1. api.aws	
		pcs.us-gov-east-1. api.aws	
AWS GovCloud (美國西部)	us-gov-west-1	pcs.us-gov-west-1. amazonaws.com	HTTPS
		pcs-fips.us-gov-west-1. amazonaws.com	
		pcs-fips.us-gov-west-1. api.aws	
		pcs.us-gov-west-1. api.aws	

Service Quotas

名稱	預設	可調整	Description
叢集	5	是	每個叢集的數量上限 AWS 區域。

Note

預設值是設定的初始配額 AWS。這些預設值與實際套用的配額值和可能的服務配額上限不同。如需詳細資訊，請參閱《Service Quotas 使用者指南》中的 [Service Quotas 術語](#)。

這些服務配額列於 [AWS 平行運算服務 \(PCS\) AWS 管理主控台](#)。若要針對顯示為可調整的值請求增加配額，請參閱 Service Quotas 使用者指南中的 [請求增加配額](#)。Service Quotas

Important

請記得檢查 [AWS 區域](#) 設定 AWS 管理主控台。

內部配額

下列配額為內部且無法調整。

名稱	預設	可調整	Description
並行叢集建立	1	否	狀態中的叢集數量上限 Creating AWS 區域。
每個叢集的運算節點群組	10	否	每個叢集的運算節點群組數量上限。
每個叢集的佇列數	10	否	每個叢集的佇列數量上限。

其他服務的相關配額 AWS

AWS PCS 使用其他 AWS 服務。這些服務的服務配額會影響您使用 AWS PCS。

影響 AWS PCS 的 Amazon EC2 服務配額

- Spot 執行個體請求
- 執行隨需執行個體

- 啟動範本
- 啟動範本版本
- Amazon EC2 API 請求

如需詳細資訊，請參閱 [《Amazon Elastic Compute Cloud 使用者指南》](#) 中的 Amazon EC2 服務配額。

平行 AWS 運算服務中的問題故障診斷

下列主題提供指引，以針對您在 AWS PCS 中可能遇到的一些問題進行疑難排解。

- [叢集更新](#)
- [運算節點引導問題](#)
- [自訂 Slurm 設定](#)
- [EC2 執行個體在重新啟動後終止](#)
- [身分和存取](#)
- [任務提交 MaxJobCount 限制](#)
- [Slurm 重新啟動問題](#)

AWS PCS 中的 EC2 執行個體會在重新啟動後終止並取代

問題概觀

重新啟動運算節點群組中的 EC2 執行個體後，AWS PCS 會自動終止並取代執行個體。

為什麼會發生這種情況

AWS PCS 不支援執行個體重新啟動。如果 EC2 執行個體重新啟動，AWS PCS 會將執行個體視為運作狀態不佳，並予以取代。如果 AWS PCS 持續終止並取代您的執行個體，可能是因為執行個體啟動後重新啟動。一些範例包括在 EC2 執行個體上透過自動化重新啟動（例如修補後的自動重新啟動）、EC2 執行個体外部的自動化（例如網路管理應用程式）、其他服務 AWS（例如 AWS Systems Manager），或由人員手動重新啟動。

處理方式

您可以檢查 `slurmctld` 或 `slurmd` 日誌，查看您的執行個體是否已重新啟動。如需詳細資訊，請參閱 [AWS PCS 中的排程器日誌](#) 及 [使用 Amazon CloudWatch 監控 AWS PCS 執行個體](#)。下列範例 `slurmctld` 日誌項目表示執行個體已重新啟動：

Example

```
[2024-09-12T06:42:50.393+00:00] validate_node_specs: Node Login-1 unexpectedly rebooted  
boot_time=1726123354 last_response=1726123285
```

由於修補而重新啟動

套用修補程式後，通常需要重新啟動。請勿將修補程式直接套用至屬於 AWS PCS 運算節點群組的 EC2 執行個體。如果您必須修補 EC2 執行個體，您應該將修補程式套用至更新的 Amazon Machine Image (AMI)，並更新運算節點群組以使用更新的 AMI。AWS PCS 為這些運算節點群組啟動的新 EC2 執行個體將使用更新的（修補）AMI。如需詳細資訊，請參閱 [AWS PCS 的自訂 Amazon Machine Image AMIs](#)。

對 AWS PCS 中的運算節點引導和註冊問題進行故障診斷

當運算節點無法開機或向 AWS PCS 叢集正確註冊時，您可能會遇到下列症狀：

- 任務未啟動
- 您無法連線到 中的執行個體 AWS Systems Manager
- 執行個體意外關閉
- 執行個體會持續取代

這些故障可能是由 EC2 執行個體啟動期間或 AWS PCS 運算節點引導程序期間的問題所造成。本主題說明在 AWS PCS 節點引導程序期間協助您疑難排解問題的程序。如需有關對 EC2 執行個體啟動進行故障診斷的詳細資訊，請參閱 [《Amazon Elastic Compute Cloud 使用者指南》](#) 中的 [對 Amazon EC2 執行個體啟動問題進行故障診斷](#)。

當 EC2 執行個體成功啟動，但在加入 AWS PCS 叢集的過程中失敗時，就會發生引導失敗。引導程序包含兩個主要階段：

- 節點註冊 – EC2 執行個體會呼叫 [RegisterComputeNodeGroupInstance](#) AWS PCS API 動作，以向 AWS PCS 服務註冊。由於下列問題，可能會發生失敗：
 - 許可
 - [錯誤的執行個體描述檔](#)
 - 聯網
 - [無法連線至 AWS PCS 端點](#)
 - [設定錯誤的 AWS PCS 端點](#)
 - [沒有公有 IP 的公有子網路中的執行個體](#)
 - [公有子網路中的多 NIC 執行個體](#)
 - 叢集秘密
 - [叢集秘密已刪除或標記為刪除](#)
- Slurm 整合 – 執行個體會執行 `slurmd` 並加入 Slurm 叢集。由於下列問題，可能會發生失敗：

- 許可
 - [安全群組組態](#)
 - [Slurmctld 無法 ping 運算節點](#)
- 自訂 AMI 設定
 - [缺少 NVIDIA 驅動程式](#)
 - [已達到 ResumeTimeout](#)

Slurm 如何在 AWS PCS 上運作

這可能有助於您將標準 Slurm 的運作方式與 Slurm 在 AWS PCS 上的運作方式進行比較。

標準 Slurm 任務處理

標準 Slurm 任務處理中會發生下列步驟：

1. 當您提交任務時，會slurmctld驗證任務並排入佇列。
2. 當資源可用時，會slurmctld配置現有的節點。
3. slurmd 協助程式在已配置節點上執行任務。

AWS PCS 上的 Slurm 任務處理

AWS PCS 任務處理中會發生下列步驟：

1. 當您提交任務時，會slurmctld驗證任務並排入佇列。
2. 當需要額外容量時，AWS PCS 會使用運算節點群組的啟動範本來啟動新的 EC2 執行個體。
3. 叢集中的新執行個體引導：
 - a. 執行個體向 AWS PCS 註冊。
 - b. 執行個體會加入 Slurm 叢集。
4. 當資源準備就緒時，會slurmctld配置節點（包括新啟動的節點）。
5. slurmd 協助程式在已配置節點上執行任務。

擷取執行個體日誌

疑難排解運算節點引導問題的第一步是擷取執行個體日誌。您可以使用下列其中一種方法：

AWS CLI

使用下列命令從運算節點擷取主控台輸出：

```
aws ec2 get-console-output --region us-east-1 --instance-id i-1234567890abcdef0 --output text
```

將 *us-east-1* 取代為您的 AWS 區域，並將 *i-1234567890abcdef0* 取代為您的執行個體 ID。

AWS Systems Manager

如果您可以使用 Systems Manager 連線到執行個體，您可以直接檢視引導日誌檔案：

1. 使用 Systems Manager 連線至執行個體。如需詳細資訊，請參閱 Systems Manager 使用者指南中的 [啟動工作階段](#)。
2. 檢視引導日誌檔案：

```
sudo cat /var/log/amazon/pcs/bootstrap.log
```

Note

如果在初始化階段發生問題，您可能需要等待約 20 分鐘，才能連線到執行個體。Systems Manager 和 SSH 服務只有在初始化完成後，或在發生故障時引導執行達到逾時時才會啟動。

從執行個體 ID 擷取 VPC/Subnet/Security 群組

若要疑難排解運算節點的問題，您可能需要擷取與執行個體相關聯之 VPC、子網路和安全群組的相關資訊。如果您不知道執行個體 IDs，請參閱 [尋找 AWS PCS 中的運算節點群組執行個體](#)。

AWS 管理主控台

取得 VPC、子網路和安全群組

1. 開啟 [Amazon EC2 主控台](#)。
2. 選擇 Instances (執行個體)。
3. 在執行個體表格中，選擇執行個體 ID。
4. 在執行個體的顯示執行個體摘要中尋找 VPC ID 和子網路 ID。

5. 在執行個體摘要中，選擇安全索引標籤。
6. 在安全索引標籤中尋找安全群組。

AWS CLI

使用下列命令來擷取執行個體的 VPC、子網路和安全群組資訊：

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query  
'Reservations[*].Instances[*].  
{InstanceId:InstanceId,VpcId:VpcId,SubnetId:SubnetId,SecurityGroups:SecurityGroups[*]}.GroupI  
--output table
```

節點註冊問題

節點註冊是運算節點在引導期間執行的第一個動作。節點會呼叫 AWS PCS API 端點來向 AWS PCS 註冊。註冊失敗通常會顯示類似以下的錯誤訊息：

```
<13>Nov 13 16:23:50 user-data: [2025-11-13T16:23:50.510+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Registering node to cluster <clusterId>  
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.192+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Retriable exception detected.  
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.193+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Response is [specific error message]  
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.194+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Retrying in 31 seconds...  
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.192+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Retriable exception detected.  
...  
<13>Nov 13 16:25:18 user-data: [2025-11-13T16:25:18.195+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Registration timeout (600 seconds) reached. Exiting.  
<13>Nov 13 16:25:18 user-data: [2025-11-13T16:25:18.200+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: ERROR: Error: (2) occurred on line 1 when running /opt/aws/pcs/  
bin/pcs_bootstrap_init.sh. Shutting down instance.
```

錯誤的執行個體描述檔

如果節點因為執行個體描述檔錯誤而無法註冊，您會看到下列錯誤：

```
<13>Nov 13 18:43:08 user-data: [2025-11-13T18:43:08.268+00:00] - /opt/aws/pcs/bin/  
pcs_bootstrap_init.sh: INFO: Response is {
```

```
<13>Nov 13 18:43:08 user-data:  "__type":  
  "com.amazon.coral.service#AccessDeniedException",  
<13>Nov 13 18:43:08 user-data:  "Message": "User: arn:aws:sts::<accountId>:assumed-  
role/<roleName>/<instanceId> is not authorized to perform:  
pcs:RegisterComputeNodeGroupInstance on resource:  
arn:aws:pcs:<regionCode>:<accountId>:cluster/<clusterId> as either the resource does  
not exist, some policy explicitly denies access, or no policy grants access",  
<13>Nov 13 18:43:08 user-data:  "nodeID": null  
<13>Nov 13 18:43:08 user-data: }
```

確認與運算節點相關聯的執行個體描述檔具有 `pcs:RegisterComputeNodeGroupInstance` 許可。如需如何建立有效執行個體描述檔的詳細資訊，請參閱 [建立 AWS PCS 的執行個體描述檔](#)。

無法連線至 AWS PCS 端點

如果您的運算節點位於私有子網路中，請確定您已為 AWS PCS 設定 VPC 端點，或子網路具有路由至 NAT 閘道以進行網際網路存取。如需詳細資訊，請參閱下列內容：

- 《Amazon Virtual Private Cloud AWS PrivateLink 指南》中的 [使用介面 VPC 端點存取 AWS 服務](#)。
- [AWS PCS 的端點和服務配額](#)。
- 《Amazon Virtual Private Cloud 使用者指南》中的 [將您的 VPC 連線到其他網路](#)
- [AWS PCS 網路](#)

設定錯誤的 AWS PCS 端點

如果您看到類似以下的錯誤訊息，請確認與您的 AWS PCS VPC 端點相關聯的政策：

```
com.amazon.coral.security.AccessDeniedException: User: arn:aws:sts::xxx:assumed-  
role/<roleName>/<instanceId> is not authorized to perform:  
pcs:RegisterComputeNodeGroupInstance on resource:  
arn:aws:pcs:<regionCode>:<accountId>:cluster/<clusterId> as either the resource does  
not exist, some policy explicitly denies access, or no policy grants access
```

如需如何為 AWS PCS 設定 VPC 介面端點的詳細資訊，請參閱 [AWS Parallel Computing 服務 使用界面端點存取 \(AWS PrivateLink\)](#)。

沒有公有 IP 的公有子網路中的執行個體

如果您的子網路未啟用自動指派公有 IP，且您的路由組態使用網際網路閘道，則執行個體無法與 AWS PCS API 通訊。

子網路中具有網際網路閘道的執行個體必須具有公有 IP 地址。若要解決此問題，請選擇下列其中一個選項：

- 將 AWS PCS 的 VPC 端點新增至叢集 VPC。這可讓執行個體與 AWS PCS 通訊，而不需要公有 IP 地址即可通過網際網路閘道。
- 使用具有 NAT 閘道的私有子網路，因此不需要公有 IP 地址。
- 透過子網路或啟動範本啟用自動公有 IP 地址指派，讓執行個體可以透過網際網路閘道聯絡 API。請注意，此選項不適用於多網路介面執行個體。

公有子網路中的多 NIC 執行個體

如果您使用的執行個體類型具有多個網路介面 (NICs)，則必須使用私有子網路。

AWS 公有 IP 地址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](#)。

多 NIC 執行個體類型需要子網路中的 NAT 閘道或內部代理才能存取 AWS PCS 端點。或者，您可以將 AWS PCS 的 VPC 端點新增至叢集 VPC。

叢集秘密已刪除或標記為刪除

如果 AWS Secrets Manager 中的 Slurm 共用秘密已刪除或標記為刪除，則運算節點將無法註冊，且您的叢集將會受損。

AWS 當您建立叢集時，PCS 會自動在 AWS Secrets Manager 中建立 Slurm 共用秘密（名稱格式為：`pcs!slurm-secret-<cluster-id>`）。叢集中的安全通訊需要此秘密。如需詳細資訊，請參閱在[AWS PCS 中使用叢集秘密](#)。

如果此秘密已刪除或標記為刪除，新節點將無法加入叢集，且控制器或其他叢集協助程式（例如 `slurmd` 和 `slurmdbd`）在重新啟動時可能無法重新加入叢集。

若要解決此問題，如果已刪除的秘密仍在復原時段內，您可以還原該秘密。如需詳細說明，請參閱[還原 AWS Secrets Manager 秘密](#)。

如果復原時段過期，則無法還原秘密，也無法還原受影響的 AWS PCS 叢集。您需要建立具有相同組態的新叢集。AWS PCS 會自動建立新的排程器秘密。

Slurm 叢集聯結問題

節點註冊成功後，運算節點會嘗試加入 Slurm 叢集。節點上的 `slurmd` 協助程式會聯絡 Slurm 控制器以向叢集註冊。Slurm 聯結失敗通常會顯示類似以下的錯誤訊息：

```
<13>Nov  5 17:20:29 user-data: [2024-11-05T17:20:28+00:00] FATAL:
Mixlib::ShellOut::ShellCommandFailed: service[slurmd] (aws-pcs-slurm::finalize_slurm
line 18) had an error: Mixlib::ShellOut::ShellCommandFailed: Expected process to exit
with [0], but received '1'
<13>Nov  5 17:20:29 user-data: ---- Begin output of ["/usr/bin/systemctl", "--system",
"start", "slurmd"] ----
<13>Nov  5 17:20:29 user-data: STDOUT:
<13>Nov  5 17:20:29 user-data: STDERR: Job for slurmd.service failed because the
control process exited with error code. See "systemctl status slurmd.service" and
"journalctl -xe" for details.
<13>Nov  5 17:20:29 user-data: ---- End output of ["/usr/bin/systemctl", "--system",
"start", "slurmd"] ----
```

安全群組組態

確認您的安全群組已正確設定，以允許運算節點和 Slurm 控制器之間的通訊。安全群組必須允許下列流量：

- slurmd 用於與 通訊的連接埠 6817 slurmctld
- slurmctld 用於 ping 的連接埠 6818 slurmd

如需安全群組需求的詳細資訊，請參閱下列主題：

- [建立 AWS PCS 的安全群組](#)
- [建立 AWS PCS 的啟動範本](#)
- [安全群組需求和考量事項](#)

Important

您在叢集建立期間與叢集相關聯的叢集安全群組也必須在運算節點群組安全群組中設定，以允許運算節點與控制器通訊。

缺少 NVIDIA 驅動程式

如果執行個體引導正確，但任務未啟動，而且您在執行個體日誌中看到類似以下的錯誤訊息，您可能缺少 NVIDIA 驅動程式：

```
<13>Dec  2 13:52:00 user-data: [2024-12-02T13:52:00.094+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_config_always.sh: INFO: nvidia-smi not found!
...
<13>Dec  2 13:54:10 user-data: Job for slurmd.service failed because the control
process exited with error code. See "systemctl status slurmd.service" and "journalctl
-xe" for details.
<13>Dec  2 13:54:12 user-data: [2024-12-02T13:54:12.718+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_finalize.sh: INFO: systemctl could not start slurmd!
```

如果您連線到執行個體並檢查slurmd協助程式狀態，您可能會看到類似以下的錯誤：

```
$ systemctl status slurmd
...
fatal: can't stat gres.conf file /dev/nvidia0: No such file or directory
```

若要解決此問題，請在您的自訂 AMI 上安裝 NVIDIA 驅動程式。如需詳細資訊，請參閱[步驟 4 – \(選用\) 安裝其他驅動程式、程式庫和應用程式軟體](#)。

已達到 ResumeTimeout

如果運算節點及其 EC2 執行個體因節點運作狀態不佳而終止，AWS PCS 可能不支援 AMI 或可能有網路問題。EC2 執行個體會執行約 30 分鐘，直到達到 Slurm 的 ResumeTimeout，並將節點標記為 DOWN。

如果執行個體未正確引導，且未向 AWS PCS 註冊（不 RegisterComputeNodeGroupInstance 呼叫 EC2 執行個體），請檢查您的執行個體日誌是否有類似以下的錯誤訊息：

```
/opt/aws/pcs/bin/pcs_bootstrap_init.sh: No such file or directory
```

此錯誤表示 AWS PCS 引導軟體不屬於 AMI。若要解決此問題，請確定您的自訂 AMI 包含 AWS PCS 引導軟體。如需詳細資訊，請參閱[AWS PCS 的自訂 Amazon Machine Image AMIs](#)。

Slurmctld 無法 ping 運算節點

如果執行個體正確執行引導程序，並已向 AWS PCS 註冊，但 slurmctld 無法看到它並向其提交任務，執行個體會有一段時間 DOWN 後設定為，然後終止。

這可能是因為設定錯誤的安全群組所造成。例如，如果啟用連接埠 6817 以允許與 slurmd 通訊 slurmctld，但連接埠 6818 遺失以允許 slurmctld ping slurmd。

確認您的安全群組包含所有必要的規則，如 中所述[安全群組需求和考量事項](#)。

故障診斷因 MaxJobCount 限制而導致的任務提交失敗

問題：任務提交失敗，並顯示下列錯誤訊息：

```
sbatch: error: Slurm temporarily unable to accept job, sleeping and retrying
```

即使執行中和待定任務的數量似乎遠低於叢集的任務限制，也會發生此錯誤。

原因：MaxJobCount 限制包括 Slurm 追蹤的所有任務，而不只是執行中或待定任務。已完成的任務會保留在 Slurm 的記憶體中一段時間（預設為 5 分鐘），然後再進行清除。在高任務輸送量期間，作用中和最近完成的任務總數可能會超過限制。

您可以在叢集節點上執行下列命令來驗證任務總數：

```
scontrol show jobs | grep -c JobId
```

這會顯示 Slurm 正在追蹤的任務總數，包括等待清除的已完成任務。

解決方案：考慮下列其中一種方法：

- 建立較大的叢集 – 如果您的工作負載持續需要更多並行任務，請建立大小較大的新叢集。如需叢集大小及其限制的詳細資訊，請參閱 [AWS PCS 中的叢集大小](#)。
- 降低任務提交率 – 調整任務提交指令碼，以較慢的速度提交任務，以便從 Slurm 追蹤中清除完成的任務時間。

AWS PCS 使用者指南的文件歷史記錄

下表說明 AWS PCS 文件的重要變更。

Date	變更	說明文件更新	API 版本已更新
2026 年 4 月 16 日	AWS 歐洲 (米蘭) 發行的 PCS	<p>AWS PCS 現已在歐洲 (米蘭) (eu-south-1) 推出。</p> <p>CloudFormation 範本可在歐洲 (米蘭) 開始使用 AWS 區域。如需詳細資訊，請參閱使用 CloudFormation 建立範例 AWS PCS 叢集及CloudFormation 建立範例 AWS PCS 叢集的範本。</p>	N/A
2026 年 3 月 10 日	更新的 PCS 代理程式	<p>已更新 AWS PCS 代理程式 1.3.2-1 的 AMI 主題。修正影響 RHEL 8.10 和 Rocky Linux 8.10 運算節點引導的問題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式及AWS PCS 代理程式版本。</p>	N/A
2026 年 2 月 11 日	AWS 在亞太區域 (孟買) 和歐洲 (巴黎) 發行的 PCS	<p>AWS PCS 現已在亞太區域 (孟買) (ap-south-1) 和歐洲 (巴黎) (eu-west-3) 提供。</p> <p>CloudFormation 範本可在亞太區域 (孟買) AWS 區域 和歐洲 (巴</p>	N/A

Date	變更	說明文件更新	API 版本已更新
		<p>黎) 開始使用 AWS 區域。如需詳細資訊，請參閱使用 CloudFormation 建立範例 AWS PCS 叢集及CloudFormation 建立範例 AWS PCS 叢集的範本。</p>	
2025 年 11 月 18 日	新功能：Slurm REST API	Slurm 25.05 或更新版本現在支援 Slurm REST API。如需詳細資訊，請參閱 AWS PCS 中的 Slurm REST API 。	AWS 開發套件：2025-11-18
2025 年 11 月 17 日	已更新 Slurm 安裝程式	更新 AWS PCS Slurm 安裝程式 24.11.7-1 和 25.05.5-1 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 及 AWS PCS 代理程式版本 。	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 11 月 10 日	新功能：支援 Slurm CLI 篩選條件外掛程式	AWS PCS 現在支援 Slurm CLI 篩選條件外掛程式執行自訂 Lua 指令碼，在任務提交參數到達 Slurm 控制器之前對其進行驗證和修改。使用 CLI 篩選條件強制執行自訂政策、設定預設參數，並在任務提交期間提供使用者指引。此功能需要 Slurm 25.05 版或更新版本。如需詳細資訊，請參閱 使用 Slurm CLI 篩選條件外掛程式在 AWS PCS 中自訂任務提交 。	N/A
2025 年 11 月 7 日	更新的 PCS 代理程式	已更新 AWS PCS 代理程式 1.3.1-1 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式及 AWS PCS 代理程式版本 。	N/A
2025 年 11 月 3 日	更新的 PCS 代理程式和 Slurm 安裝程式	更新 AWS PCS 代理程式 1.3.0-1 和 Slurm 安裝程式 24.11.6-2、24.05.8-2 和 23.11.10-4 的 AMI 主題。更新支援的作業系統清單。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式及 AWS PCS 代理程式版本 。	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 10 月 23 日	更新內容： https://pcs-multi-cluster-login-configure.sh	修正多叢集登入節點組態指令碼中的一些錯誤。如需詳細資訊，請參閱 AWS PCS 多叢集登入節點組態指令碼程式碼 。	N/A
2025 年 10 月 21 日	新功能：叢集秘密輪換	AWS PCS 現在支援叢集秘密輪換，以增強安全性。如需詳細資訊，請參閱 在 AWS PCS 中輪換叢集秘密 。 更新支援叢集秘密輪換的最低管理員許可。如需詳細資訊，請參閱 AWS PCS 的最低許可 。	N/A
2025 年 10 月 17 日	新主題：多叢集登入節點組態指令碼	新增了新主題，提供指令碼來設定獨立登入節點以連線至多個 AWS PCS 叢集。指令碼會自動設定多個 Slurm sackd 協助程式，並建立叢集互動的啟用指令碼。 如需詳細資訊，請參閱 將獨立登入節點連接到 AWS PCS 中的多個叢集 。	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 10 月 16 日	針對 Slurm 25.05 更新	<p>已更新 Slurm 25.05 支援的使用者指南。Slurm 25.05 現在是預設版本。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none">• AWS PCS 中的 Slurm 版本• 建置 AWS PCS 自訂 AMIs 的軟體安裝程式• AWS PCS 範例 AMIs 版本備註	N/A
2025 年 10 月 16 日	更新的 PCS 代理程式	<p>更新 AWS PCS 代理程式 1.2.2-1 的 AMI 主題。如需詳細資訊，請參閱建置 AWS PCS 自訂 AMIs 的軟體安裝程式及AWS PCS 代理程式版本。</p>	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 10 月 2 日	新功能：Slurm 節點重新啟動、叢集更新和自訂 Slurm 設定	<p>AWS PCS 新增對多個新功能的支援：</p> <ul style="list-style-type: none"> • Slurm 節點重新啟動 – 使用 Slurm 的原生 <code>scontrol reboot</code> 命令重新啟動運算節點，無需替換執行個體。如需詳細資訊，請參閱在 AWS PCS 中使用 Slurm 重新啟動運算節點。 • 叢集更新 – 在建立後修改叢集組態，無需重建。如需詳細資訊，請參閱在 AWS PCS 中更新叢集。 • Slurm 自訂設定 – 跨叢集、佇列和運算節點群組資源設定進階 Slurm 參數。如需詳細資訊，請參閱在 AWS PCS 中設定自訂 Slurm 設定。 	2025-10-01
2025 年 9 月 23 日	新的疑難排解主題：運算節點引導問題	<p>新增診斷和解決運算節點引導問題的疑難排解指引。如需詳細資訊，請參閱對 AWS PCS 中的運算節點引導和註冊問題進行故障診斷。</p>	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 9 月 17 日	新功能：ML 的容量區塊	<p>AWS PCS 現在支援適用於 ML 的 Amazon EC2 容量區塊，可讓您為叢集保留 GPU 加速運算執行個體。如需詳細資訊，請參閱搭配 AWS PCS 使用適用於 ML 的 Amazon EC2 容量區塊。</p> <p>支援容量區塊的最低許可現在是服務管理員的最低許可的一部分。如需詳細資訊，請參閱AWS PCS 的最低許可。</p>	2025-09-17
2025 年 9 月 11 日	AWS 受管政策更新	<p>AWS PCS 已更新 AWSPCSServiceRolePolicy 以支援容量區塊。如需詳細資訊，請參閱AWS AWS Parallel Computing Service 的受管政策。</p>	N/A
2025 年 8 月 14 日	已更新執行個體描述檔文件	<p>使用建立 IAM 角色和執行個體描述檔的完整 CLI 說明來增強執行個體描述檔文件。新增使用 設定執行個體描述檔step-by-step程序，AWS CLI 以及尋找與 AWS PCS 搭配使用之執行個體描述檔的改善指引。</p> <p>如需詳細資訊，請參閱平行運算服務的 IAM AWS 執行個體描述檔。</p>	2025-08-14

Date	變更	說明文件更新	API 版本已更新
2025 年 8 月 1 日	新主題：SPANK 外掛程式	<p>新增了 SPANK（適用於節點和任務 Kontrol 的 Slurm 外掛程式架構）外掛程式的文件，可用於在 AWS PCS 叢集上啟動和執行任務期間擴展和修改 Slurm 的行為。</p> <p>如需詳細資訊，請參閱 使用 SPANK 外掛程式擴展 AWS PCS 上的 Slurm 功能。</p>	N/A
2025 年 8 月 1 日	IPv6 網路支援	<p>新增在建立 AWS PCS 叢集時對 IPv6 聯網的支援。您現在可以選擇 IPv6 做為叢集的網路類型，並對應更新 VPC 需求、子網路組態、安全群組設定和叢集建立程序。</p> <p>如需詳細資訊，請參閱 AWS PCS VPC 和子網路需求和考量事項及在 AWS PCS 中建立叢集。</p>	2025-08-01

Date	變更	說明文件更新	API 版本已更新
2025 年 7 月 3 日	AWS 歐洲（倫敦）發行的 PCS	<p>AWS PCS 現已在歐洲（倫敦）(eu-west-2) 推出。</p> <p>CloudFormation 範本可在歐洲（倫敦）開始使用 AWS 區域。如需詳細資訊，請參閱使用 CloudFormation 建立範例 AWS PCS 叢集及CloudFormation 建立範例 AWS PCS 叢集的範本。</p>	N/A
2025 年 7 月 1 日	更新主控台指示	<p>您現在可以讓 AWS PCS 在主控台中建立叢集和運算節點群組時為您建立基本執行個體描述檔和安全群組。如需詳細資訊，請參閱：</p> <ul style="list-style-type: none"> • 在 AWS PCS 中建立叢集 • 在 AWS PCS 中建立運算節點群組 • 平行運算服務的 IAM AWS 執行個體描述檔 	N/A
2025 年 6 月 23 日	新的受管政策：AWSPCSComputeNodePolicy	<p>新增了新的受管政策，授予 AWS PCS 運算節點連線至 AWS PCS 叢集的許可。如需詳細資訊，請參閱AWS 受管政策：AWSPCSComputeNodePolicy。</p>	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 6 月 19 日	新主題：任務完成日誌	使用任務完成日誌，在任務完成時記錄任務的詳細資訊，無需額外費用。如需詳細資訊，請參閱 AWS PCS 中的任務完成日誌 。	N/A
2025 年 6 月 18 日	AWS 中的 PCS 版本 AWS GovCloud (US)	<p>AWS PCS 現在可在 AWS GovCloud (美國東部) (us-gov-east-1) 和 AWS GovCloud (美國西部) (us-gov-west-1) 中使用。</p> <p>CloudFormation 範本可在 中開始使用 AWS GovCloud (US) Regions。如需詳細資訊，請參閱使用 CloudFormation 建立範例 AWS PCS 叢集及 CloudFormation 建立範例 AWS PCS 叢集的範本。</p> <p>如需 中 AWS PCS 服務端點的詳細資訊 AWS GovCloud (US) Regions，請參閱 AWS PCS 的端點和服務配額。</p> <p>如需 中差異的詳細資訊 AWS GovCloud (US) Regions，請參閱 AWS GovCloud (US) 《使用者指南》AWS 中的 PCS AWS GovCloud (US)。</p>	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 6 月 18 日	更新的 PCS 代理程式	更新 AWS PCS 代理程式 1.2.1-1 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 。	N/A
2025 年 5 月 15 日	新功能：會計	Slurm 24.11 或更新版本現在支援 Slurm 會計。如需詳細資訊，請參閱 AWS PCS 中的 Slurm 會計 。	AWS 開發套件：2025-05-15
2025 年 5 月 15 日	針對 Slurm 24.11 更新	已更新 Slurm 24.11.5 支援的使用者指南。如需詳細資訊，請參閱下列內容： <ul style="list-style-type: none"> • AWS PCS 中的 Slurm 版本 • 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 • AWS PCS 範例 AMIs 版本備註 	N/A
2025 年 5 月 5 日	已更新 Slurm 版本常見問答集	更新 Slurm 版本常見問答集 (FAQ)，有關接近或超過生命週期結束 (EOL) 的 Slurm 版本。如需詳細資訊，請參閱 有關 AWS PCS 中 Slurm 版本的常見問題 。	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 4 月 17 日	新主題：如何取得運算節點群組詳細資訊	了解如何取得 AWS PCS 運算節點群組的詳細資訊，例如其 ID、ARN 和 AMI ID。如需詳細資訊，請參閱 在 AWS PCS 中取得運算節點群組詳細資訊 。	N/A
2025 年 4 月 2 日	已更新 Slurm 安裝程式	已更新 Slurm 安裝程式 24.05.7-1 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 。	N/A
2025 年 3 月 28 日	新增運算節點群組和佇列數目上限的配額	針對每個叢集的運算節點群組數目上限和每個叢集的佇列數目上限，新增了不可調整的內部配額。如需詳細資訊，請參閱 內部配額 。	N/A
2025 年 3 月 14 日	已變更 CloudFormation 範本中的屬性索引鍵	Id 現在 TemplateId 適用於 CloudFormation 範本中的 CustomLaunchTemplate 屬性。如需詳細資訊，請參閱 AWS PCS CloudFormation 範本的一部分 中的 Resources 。	N/A

Date	變更	說明文件更新	API 版本已更新
2025 年 3 月 13 日	新增 AWS PCS 代理程式和 Slurm 的版本資訊	<p>新增了新主題，說明每個版本的 AWS PCS 代理程式的變更。如需詳細資訊，請參閱AWS PCS 代理程式版本。</p> <p>已將詳細資訊新增至 Slurm 版本主題，該主題說明對 Slurm 的 AWS PCS 支援的重要支援日期和詳細版本備註。如需詳細資訊，請參閱AWS PCS 中的 Slurm 版本。</p>	N/A
2025 年 3 月 7 日	更新的 PCS 代理程式	更新 AWS PCS 代理程式 1.2.0-1 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 。	N/A
2025 年 2 月 3 日	新增使用 AWS CloudFormation 搭配 AWS PCS 的主題	已將主題新增至 使用者指南，提供如何使用 CloudFormation 搭配 AWS PCS 的範例。本主題提供使用範例 CloudFormation 範本建立範例 AWS PCS 叢集的程序，並簡短說明該範本各節。如需詳細資訊，請參閱 CloudFormation 和 AWS PCS 入門 。	N/A

Date	變更	說明文件更新	API 版本已更新
2024 年 12 月 18 日	針對 Slurm 24.05 更新	已更新 Slurm 24.05 支援的使用者指南。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 及 AWS PCS 範例 AMIs 版本備註 。	N/A
2024 年 12 月 18 日	更新 Slurm 23.11 範例 AMIs NVIDIA 版本	已更新 Slurm 23.11 範例 AMIs 中的 NVIDIA 驅動程式和 CUDA 版本。如需詳細資訊，請參閱 AWS PCS 範例 AMIs 版本備註 。	N/A
2024 年 12 月 17 日	已更新 Slurm 安裝程式	已更新 Slurm 安裝程式 23.11.10-3 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 。	N/A
2024 年 12 月 13 日	更新的 PCS 代理程式	更新 AWS PCS 代理程式 1.1.1-1 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 。	N/A
2024 年 12 月 6 日	更新的 PCS 代理程式和 Slurm 安裝程式	更新 AWS PCS 代理程式 1.1.0-1 和 Slurm 安裝程式 23.11.10-2 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式 。	N/A
2024 年 12 月 6 日	新增有關作業系統支援的主題	如需詳細資訊，請參閱 AWS PCS 中支援的作業系統 。	N/A

Date	變更	說明文件更新	API 版本已更新
2024 年 11 月 8 日	重組使用者指南	我們重組了使用者指南，將主題帶到頂層、將一些主題移至自己的頁面，並將類似的主題分組在一起。	N/A
2024 年 11 月 7 日	已更新 AMI 主題	更新 Slurm 23.11.10 和 libjwt 17.0 的 AMI 主題。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式及步驟 3 – 安裝 Slurm 。 簡化並更正 AMIs 的版本備註。如需詳細資訊，請參閱 AWS PCS 範例 AMIs 版本備註 。	N/A
2024 年 11 月 7 日	新增了有關搭配 AWS PCS 使用加密 EBS 磁碟區的新主題	新增主題，說明 AWS PCS 中加密 EBS 磁碟區所需的 KMS 金鑰政策。如需詳細資訊，請參閱在 AWS PCS 中與加密 EBS 磁碟區搭配使用所需的 KMS 金鑰政策 。	N/A
2024 年 10 月 18 日	AWS PCS 代理程式 1.0.1-1 已發行	更新 AMI 相關文件以參考 AWS PCS 代理程式 1.0.1-1 版。如需詳細資訊，請參閱 建置 AWS PCS 自訂 AMIs 的軟體安裝程式及步驟 2 – 安裝 AWS PCS 代理程式 。	N/A

Date	變更	說明文件更新	API 版本已更新
2024 年 10 月 10 日	新增故障診斷章節	新增故障診斷章節，其中包含重新啟動後自動取代之 EC2 執行個體的相關主題。如需詳細資訊，請參閱 平行 AWS 運算服務中的問題故障診斷 。	N/A
2024 年 9 月 23 日	更新使用 API 動作和服務管理員的最低許可	CreateComputeNodeGroup 和 UpdateComputeNodeGroup API 動作現在需要 ec2:DescribeInstanceTypeOfferings 許可。如需詳細資訊，請參閱 AWS PCS 的最低許可 。	N/A
2024 年 9 月 5 日	更新服務管理員最低許可的範例 IAM 政策	如需詳細資訊，請參閱 服務管理員的最低許可 。	N/A
2024 年 9 月 5 日	在受管政策頁面中，將缺少的許可新增至 JSON	這只是文件的更正。實際受管政策並未變更。如需詳細資訊，請參閱 AWS AWS Parallel Computing Service 的受管政策 。	N/A
2024 年 8 月 28 日	已新增受管政策頁面	如需詳細資訊，請參閱 AWS AWS Parallel Computing Service 的受管政策 。	N/A
2024 年 8 月 28 日	AWS PCS 版本	AWS PCS 使用者指南的初始版本。	AWS SDK : 2024-08-28

AWS 詞彙表

如需最新的 AWS 術語，請參閱 AWS 詞彙表 參考中的 [AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。